

AD-A233 651

DLA-91-P90272

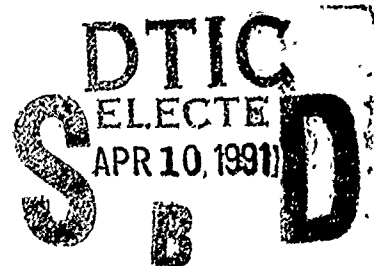
Quality Effectiveness Sensing
Technique Release 3.0

OPERATIONS RESEARCH AND ECONOMIC ANALYSIS OFFICE

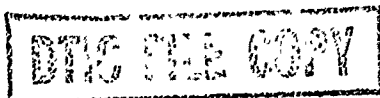


DEPARTMENT OF DEFENSE

DEFENSE LOGISTICS AGENCY



March 1991



DISTRIBUTION STATEMENT A

Approved for public release;
Distribution Unlimited

91 4 09 120

DLA-91-P90272

Quality Effectiveness Sensing Technique Release 3.0

Paul E. Grover

**DEPARTMENT OF DEFENSE
DEFENSE LOGISTICS AGENCY
OPERATIONS RESEARCH AND ECONOMIC ANALYSIS OFFICE
CAMERON STATION
ALEXANDRIA, VA 22304-6100**

March 1991



DEFENSE LOGISTICS AGENCY
HEADQUARTERS
CAMERON STATION
ALEXANDRIA, VIRGINIA 22304-6100



IN REPLY
REFER TO

DLA-LO


FOREWORD

In an ongoing effort, the Defense Logistics Agency (DLA) Operations Research and Economic Analysis Management Support Office, DLA-DORO, has developed and fielded versions of the Quality Effectiveness Sensing Technique (QUEST) Model for the Quality Assurance (QA) mission with Contract Administration Services. This report documents the logic, structure and code for QUEST release 3.0.

QUEST release 3.0 measures the effectiveness of the contractor's QA operations by comparing the contractor's key indicators with those of similar contractors (or peers). Based on those peer comparisons and trends, an effectiveness score for each indicator is computed and a weighted average of all indicators produces a bottom line effectiveness rating.

Release 3.0 was validated by incorporating the profound knowledge of experts in the weighting factors and program logic. A Study Advisory Group, consisting of field and Headquarters QA personnel guided the effort. A statistical test was passed which compared QUEST release 3.0 results with expert opinion on certain contractors.

Because of the track record of earlier releases, the use of knowledgeable experts in the development process and the results of validity tests, it is concluded that QUEST 3.0 is implementable, valid and meets the objective of measuring contractor QA effectiveness. It is recommended that release 3.0 be implemented throughout the Defense Contract Management Command (DCMC).


ROGER C. ROY
Deputy Director
Policy and Plans



Accession For	
NTIS GRA&I	<input checked="checked" type="checkbox"/>
DTIC TAB	<input type="checkbox"/>
Unannounced	<input type="checkbox"/>
Justification	
By	
Distribution/	
Availability Codes	
Dist	Avail and/or Special
A-1	

CONTENTS

<u>Title</u>	<u>Page</u>
Foreword.....	iii
Table of Contents.....	v
List of Tables.....	vii
List of Figures.....	ix
Executive Summary.....	xi
I. Introduction.....	1
A. Background.....	1
B. Problem Statement.....	1
C. Objectives.....	1
D. Scope.....	1
II. Methodology.....	1
A. General Discussion.....	1
B. Contractor Effectiveness.....	2
1. Corrective Action Requests (JAR).....	2
2. Product Quality Deficiency Reports (PQDR).....	2
3. Product Audit.....	2
4. Waivers.....	3
5. Material Review Board (MRB) Actions.....	3
6. Deviations.....	3
7. Engineering Change Proposals (ECP).....	3
8. Total Score.....	3
9. Indicator Measurement.....	3
10. Reports.....	4
III. Analysis.....	7
IV. Conclusions.....	9
V. Recommendation.....	9
VI. Benefits.....	9
VII. Implementation.....	9
Appendix A. Study Advisory Group.....	A-1
Appendix B. QUEST Source Code.....	B-1
Appendix C. References.....	C-1

LIST OF TABLES

<u>Number</u>	<u>Title</u>	<u>Page</u>
1	Sample QUEST 3.0 Report.....	5

LIST OF FIGURES

<u>Number</u>	<u>Title</u>	<u>Page</u>
1	Comparison of QUEST to Expert, Nonresident Contractors Nov 89...8	
2	Comparison of QUEST to Expert, Resident Contractors Nov 89.....8	

EXECUTIVE SUMMARY

In an ongoing effort, the Defense Logistics Agency (DLA) Operations Research and Economic Analysis Management Support Office, DLA-DORO, has developed and fielded versions of the Quality Effectiveness Sensing Technique (QUEST) Model for the Quality Assurance (QA) mission with Contract Administration Services. This report documents the logic, structure and code for QUEST release 3.0.

The major differences between release 3.0 and earlier versions are as follows. First, release 3.0 focuses exclusively on contractor QA effectiveness and has no measures for in-house program effectiveness. Secondly, this version is based on the new In-Plant Quality Evaluation Program rather than its predecessor, Contract Quality Assurance Program. Finally, the indicators used by release 3.0 are those currently available in the revised Quality Assurance Management Information System.

QUEST 3.0 computes measures of effectiveness using an analytical technique called Technique for Order Preference by Similarity to Ideal Solution (TOPSIS). TOPSIS was used in earlier releases to compute a "Product" Score, evaluating the potential for nonconforming product. TOPSIS produces percentage scores on a scale of 0-100 percent for seven key indicators by comparing a contractor's QA data with that of similar contractors and also by computing trends. Based on this peer comparison and evidence of improvement (or lack thereof), the contractor is measured and reports are generated to QA supervisors and managers.

Release 3.0 was validated by incorporating the profound knowledge of experts in the weighting factors and program logic. A Study Advisory Group, consisting of field and Headquarters QA personnel guided the effort. A statistical test was passed which compared QUEST release 3.0 results with expert opinion on certain contractors.

Because of the track record of earlier releases, the use of knowledgeable experts in the development process and the validation test results, it is concluded that QUEST 3.0 is implementable, valid and meets the objective of measuring contractor QA effectiveness. It is recommended that release 3.0 be implemented throughout the Defense Contract Management Command (DCMC).

I. INTRODUCTION

A. Background. Quality Assurance Effectiveness Sensing Technique (QUEST) was developed to provide supervisors and upper management a tool to measure and monitor the effectiveness of in-plant quality operations at Defense Contract Administration Services (DCAS). Developed in the Fiscal Year (FY) 1987 - 1988 timeframe, QUEST was implemented in FY 89. An improved version of QUEST, release 2.0, was implemented in March 1990. These versions of QUEST were based on DLAM 8200.1, Contract Quality Assurance Program or CQAP. In FY 90, a transition from CQAP to a new program began within DCAS. As the new program, called In-Plant Quality Evaluation (IQUE), phased in, QUEST 2.0 has become obsolete, requiring a new version of the model.

B. Problem Statement

1. To develop a plan to modify QUEST 2.0 to be compatible with the new policies and procedures under IQUE.

2. To establish necessary data requirements to support a new release of QUEST.

3. To develop a new release of QUEST to satisfy Defense Logistics Agency (DLA) commitments to the Department of Defense (DoD) to establish measures of effectiveness for Quality Assurance (QA).

C. Objectives

1. To develop a workable, valid model that measures QA effectiveness under IQUE.

2. To influence the functional description of the QA Management Information System (QAMIS) to assure necessary data is available to measure effectiveness.

D. Scope. QUEST 3.0 will apply to all active contractors under the surveillance of the Defense Contract Management Command (DCMC, formerly DCAS). QUEST release 3.0 will be limited to providing measures of contractor QA effectiveness only. Measures relating to the effectiveness of the government QA program will be addressed in this report but deferred for future decisions.

II. METHODOLOGY

A. General Discussion

QUEST 2.0 contained two primary effectiveness measures, namely a program score and a product score. Program scores were designed to measure the effectiveness of CQAP operations and were a report card on the government surveillance system. Product scores were designed to measure product conformance to requirements and were a report card on the contractor.

Under the IQUE philosophy, the Government and the contractor work more as a team, less as adversaries. As such, it is increasingly difficult to measure each independently. The goal of IQUE is to place less emphasis on inspecting final product and to increased emphasis on controlling production processes. Since each contractor has a unique set of processes, IQUE is very flexible and tailored to the contractor by the government Quality Assurance Representative (QAR). The regimented, standardized approach of CQAP lent itself more to measurement. IQUE will be more difficult to measure from a program perspective.

These issues were discussed by the QUEST 3.0 Study Advisory Group (SAG) (Appendix A). Because of time constraints and the perceived difficulty of developing program measures for IQUE, it was decided that release 3.0 should be limited to measuring contractor effectiveness. If the IQUE program is successful and hence effective, the results will be evident from contractor effectiveness scores. Continuous improvement of contractor effectiveness measures is a signal of the effectiveness of IQUE. The SAG felt that measures of IQUE program effectiveness should not be terminated, only given lower priority than contractor effectiveness. The group recommended that program effectiveness be considered as an add-on at a later date, if feasible.

B. Contractor Effectiveness. Like earlier versions of QUEST, release 3.0 is based on the premise that available data only signals instances of ineffectiveness. Thus indicators are called negative indicators. The more negative indicators associated with a particular contractor, the less effective that contractor is perceived to be. QUEST 3.0 will capture data on negative indicators of effectiveness and translate that data to measures of effectiveness. The indicators, currently available in the QAMIS, used by QUEST 3.0 are:

1. Corrective Action Requests (CAR). Verbal, written and escalation CARs (methods C, D, and E) are weighted and added to produce an indicator value for CARs. Relative to a method C CAR, the sum of verbal and written CARs are weighted 4 to 1 (4 verbal or written CARs count the same as a method C). Two method C CARs have equal weight to a method D and a method E has equal weight to method C. Weighting factors were determined by the SAG.

2. Product Quality Deficiency Reports (PQDR). PQDR indicators are used exactly the same way that earlier versions of QUEST evaluated Materiel Deficiency Reports, with two exceptions. PQDRs charged against an Indefinite Delivery Type Contract or Basic Ordering Agreement (Procurement Instrument Identification Number Type "D" or "G" contracts) are not discounted for age and carry full weight. Also PQDRs with defect code "X," contractor refuses to investigate, are counted as valid PQDRs. For additional information on PQDR processing, see Appendix C within Reference 1.

3. Product Audit. The ratio of product audit counts that resulted in CARs to total product audit counts is converted to a percentage value. This indicator represents the percentage of product audits that contain contractual nonconformances.

4. Waivers. The number of waivers submitted to the QAR and reported in the QAMIS is the fourth indicator. Waivers are after the fact requests to the Contracting Officer to modify or eliminate contractual requirements.

5. Material Review Board (MRB) Actions. The number of MRB actions reported by the QAR is an indicator that has not been modified from earlier versions of QUEST. MRB actions are minor nonconformances reported by the contractor.

6. Deviations. The number of deviations submitted to the QAR and reported monthly in the QAMIS is the sixth indicator. Earlier versions of QAMIS reported combined waivers and deviations. With the new QAMIS, these data elements are reported separately and are used separately in QUEST 3.0. Deviations are before-the-fact requests to deviate from contractual requirements (i.e. after the contract is awarded but before actual production of the item).

7. Engineering Change Proposals (ECP). ECPs submitted to the QAR are the last indicator used to evaluate contractor effectiveness. This indicator remains unchanged from earlier versions of QUEST. ECPs are formal requests by the contractor to permanently change the specifications.

8. Total Score. A composite total score is generated in QUEST by taking a weighted average of the preceding seven indicators, listed in the order of decreasing weight. Weights were generated by the SAG, reflecting the relative importance of each indicator in viewing contractor quality performance. Details are shown in Appendix B, page B-33.

9. Indicator Measurement

Translation from raw indicator data to a consistent measure of effectiveness is done using the Technique for Order Preference by Similarity to Ideal Solution (TOPSIS), developed by Dr. Ching-Lai Hwang of Kansas State University [2, 3]. TOPSIS is used to produce scores on a scale of 0-100 percent from multiple criteria by measuring each criteria from a set of "ideal" and "negative ideal" points. "Ideal" conditions are defined as the best possible values for each criteria or attribute. "Negative ideal" conditions are the worst possible values for each parameter or indicator. Ideal and negative ideal conditions were defined by the SAG. Definitions for ideal/negative ideal vary depending on whether the contractor is considered to be "normal" or a "problem" contractor.

A "problem" contractor is considered to be a contractor that has a history of chronic performance problems. QUEST assigns additional emphasis to the trend component of the measurement process for these contractors. To be viewed by QUEST 3.0 as a problem contractor, two of the following conditions must apply. First, the contractor is on the Contractor Alert List (for any reason). The Alert List is compiled by DCMC to warn acquisition elements of contract administration problems experienced.

Second, the contractor has a history of relatively high incidence of valid PQDRs. Third, the contractor received a method C CAR within the last three months. Fourth, the contractor received a method D Corrective Action Request within the last year. Finally, the contractor received a method E CAR within the last three months.

The Trend component in the measure of effectiveness carries half the weight in overall indicator measurement for a normal contractor and 10 percent of the weight for a problem contractor. Trend is calculated based on the last six months of activity. Activity is defined as any month when more than two hours of total government QA surveillance takes place. If the slope of the trend line is upward or positive, QUEST views this as unfavorable and penalizes the contractor. Downward or negative trends result in high effectiveness scores.

The second part of the measurement process is to measure the contractor's data in relation to the contractor's "peers." If the contractor's indicators are higher than those of "peers," QUEST produces low effectiveness scores. Lower-than-peers indicators result in high effectiveness scores. Additional information on the Peer Processing is in Appendix F of Reference 4. In all cases, except PQDR for problem contractors, ideal conditions are minus three standard deviations for trend and peer comparison. Negative ideal conditions are plus three standard deviations. A +3 standard deviation means that the trend or peer comparison is three standard deviations above average. A -3 standard deviation indicates that the contractor's trend or peer comparison is three standard deviations below average. Details are found in Appendix B, page B-33.

TOPSIS mathematically converts trend and peer comparison results into a single measure by computing two distances. A given contractor's trend is compared to ideal and negative ideal trend. Also, the indicator value relative to peers for the contractor is compared to the ideal and negative ideal conditions. The difference between the contractor's actual trend and peer comparison to ideal points is called a distance. TOPSIS combines the distances into a ratio of a part to a whole, forming a score between 0 percent and 100 percent. Specifically, TOPSIS scores are the ratio of distance from negative ideal to the total distance from both positive and negative ideal. The resulting scores are such that low scores indicate proximity to negative ideal (poor effectiveness) and high ratios indicate closeness to ideal conditions (high effectiveness).

10. Reports

QUEST 3.0, like QUEST 2.0, provides data to users in three ways. A district level report is generated to a printer. This report contains a one page summary for each unique organization (section) in the district each month. An example is shown in Table 1. Secondly, through the Mechanization of Reports Distribution System (MORDS), each division receives the same hard copy report for the organizations within the division. For users that have terminals and access to the district's DMINS system, QUEST data is electronically available in a menu-driven format.

Table 1

SAMPLE QUEST 3.0 REPORT

11 89 QUEST REPORT FOR SECTION ABC

CAGE	NAME	QAR	COMM/	QAS	CA	PQDR	CONTRACTOR EFFECTIVENESS	PA	WFRS	MRB	DEVN	ECP	TOTAL SCORE	FIRST PRIOR MONTH	SECOND PRIOR MONTH	THIRD PRIOR MONTH	PEER GRP/RTG
53670	HUGHES AIR L49 L5 A	1	54.7	100.0	100.0	55.4	100.0	48.7	69.9	64.3	71.4	56.1	75.5	51.9	60.8	75.5	755 D
ORPNO	ENERGY CON LTJ M3 A	1	33.7	100.0	100.0	50.0	100.0	81.2	50.0	100.0	100.0	50.0	72.0	51.9	60.8	75.5	803 D
86831	KAISER ELE LOJ X1 A	1	22.4	100.0	100.0	100.0	100.0	100.0	100.0	50.0	81.4	77.0	81.6	84.4	84.4	84.1	926 C
50832	RAYTHEON S LRL K1 B	2	100.0	100.0	100.0	100.0	100.0	100.0	100.0	49.1	81.6	84.1	83.9	100.0	100.0	100.0	726 C
62458	HUGHES AIR LN7 L2 C	1	18.7	100.0	100.0	100.0	100.0	100.0	100.0	100.0	83.9	87.0	87.0	84.1	84.1	82.7	786 C
55267	HUGHES AIR LFR L4 A	1	34.4	100.0	100.0	100.0	100.0	100.0	100.0	100.0	87.0	87.0	87.0	84.1	84.1	82.7	754 C
44523	DELPHI COM LKG L2 B	0	13.1	100.0	100.0	3.4	100.0	100.0	0.0	100.0	59.2	100.0	59.2	90.9	100.0	100.0	197 F
65630	SIGNAL DES LKG L2 B	0	37.8	100.0	100.0	11.1	100.0	100.0	16.7	100.0	66.6	86.4	66.6	78.0	86.4	86.4	197 F
95411	ENDEVCO LKG L2 A	0	41.3	100.0	100.0	74.6	100.0	100.0	17.0	100.0	74.8	87.2	74.8	87.2	87.2	87.2	196 D
1W195	GAM RAD WE LKG N2 B	0	100.0	100.0	100.0	39.7	100.0	100.0	21.9	100.0	84.1	82.9	84.1	87.8	87.8	82.9	245 C
96182	EATON CORP L9V L5 B	0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	206 A
45436	PILKINGTON L9V K1 B	0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	170 A
97525	ECCO INC LSU L5 C	0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	207 A
04423	TELEONIC BE LKG L5 B	0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	206 A
68615	INFOTEC DE LOZ L5 B	0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	206 A
43715	TRONTECH I L3B L5 B	0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	206 A
89178	STANCIL CO LOZ L5 C	0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	206 A
13979	PARKO ELEC L3B L5 C	0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	206 A
71279	INTERCONNE LSU E1 B	0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	207 A
41820	INTEGRATED L3B L3 B	0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	207 A
14552	MICROSEMI LOZ L2 B	0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	98 A
33434	HELIONETIC LSU L5 C	0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	200 A
11534	DUNCAN ELE L9V L4 B	0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	197 A
34156	SEMICOA L3B L4 B	0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	207 A
12522	STACOSWITC LOZ E1 B	0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	203 A
55761	BALL CORP LKG L5 B	0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	98 A
18965	DYNAMIC AI LSU E5 C	0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	206 A
57810	PAN-A-LITE L9V L5 B	0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	111 A
19565	SYMBOLIC D L3B A3 B	0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	206 A
58795	SABRITEC LSU E1 B	0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	8 A
21793	RACAL-DANA LKG L3 B	0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	98 A
53669	HUGHES AIR L3B E1 B	0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	200 A
53515	ELECTRONIC LSU L5 C	0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	98 A
52414	EMERSON EL LSU L3 B	0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	207 A
52460	RW ELECTRO L9V L5 A	0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	200 A
24027	CALMONT EN LSU E1 C	0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	205 A
28427	BARCEL WIR L3B E1 B	0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	99 A
WTD AVG	ABC																98 A
RESIDENT																	
NONRESIDENT																	
COMBINED																	

The QUEST 3.0 report is similar to previous QUEST reports [1, 4]. Generally, for QUEST 3.0, sections dealing with Program Score (Red Flags) have been eliminated, and additional descriptive data on the contractor is provided. Also, historical QUEST scores are shown to ease trend analysis. The following Table 1 headings are defined.

a. CAGE. Commercial and Government Entity Code identifies the contractor. If there is an asterisk preceding the CAGE, the contractor has been labeled by QUEST as a "problem" contractor.

b. NAME. First ten characters in the name field of the contractor.

c. OAR. Identifies the government Quality Assurance Representative in charge of the government quality surveillance.

d. COMM. Commodity Code in accordance with DLAM 8200.2.

e. PVN. QA Provision Code. A represents MIL-Q-9858A or equivalent, B represents MIL-I-45208A and C is "other."

f. QAS. The number of government QA Specialists (QAS) assigned to the facility. This number is computed from the total government QA hours charged to the contractor (roughly one man month equals 149 hours) and may not be the actual number of QASs assigned to the contractor.

g. CA. Correction Action Request score.

h. PQDR. Product Quality Deficiency Report score.

i. PA. Product Audit Score.

j. WVRS. Waiver Score.

k. MRB. Materiel Review Board score.

l. DEVN. Deviation Score.

m. ECP. Engineering Change Proposal Score.

n. TOTAL SCORE. Weighted average of the previous seven indicator scores. This represents the overall effectiveness score for the contractor.

o. PRIOR MONTH. QUEST 3.0 shows up to three prior month TOTAL SCORES for each contractor.

p. PEER GRP. Peer group identification number (see Appendix F of Reference 4). Contractors that have the same peer group number generally have the same commodity, provision and have roughly the same defense contract workload.

q. PEER RTG. Converts the Total Score to a letter grade. Since not all peer groups have the same pattern of QUEST scores (nonresidents generally get higher ratings than resident facilities), the letter grade represents a "relative to peer" effectiveness rating. For example, an identical Total Score of 80 percent could translate into a B, C or D rating, depending on the peer performance.

III. ANALYSIS

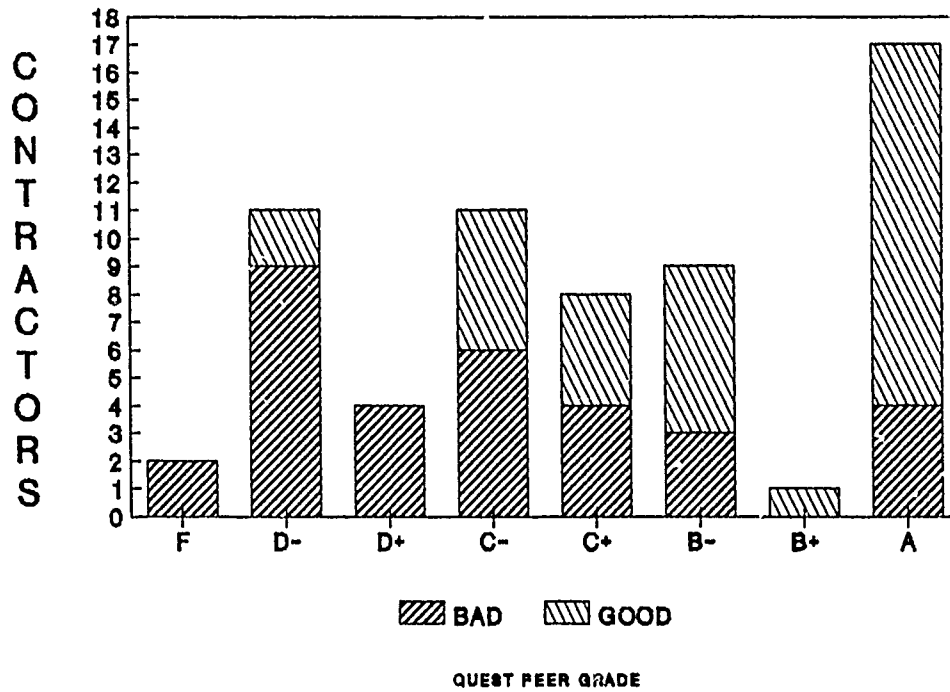
Prior to implementation, it was determined that because of the significant changes to QUEST, a revalidation was necessary. The original QUEST version was validated using a correlation technique matching QUEST scores to expert evaluations [1]. Because of the compressed schedule for QUEST 3.0, a similar but simpler nonparametric analytical process was used to determine if the QUEST scores reasonably matched the perception of experts. If the QUEST effectiveness rating correlated with expert opinion to an adequate degree, the model was considered valid by the SAG.

A Mann-Whitney rank sum test [5] was conducted. Mann-Whitney tests to determine if an effect could be caused by random coincidence or if an effect is "real." In statistical jargon, a "real" effect is called "statistically significant." We tested to see if the agreement between experts and QUEST is statistically significant. Each former DCAS region was asked to produce a set of contractors labeled as effective or ineffective from a QA viewpoint. These "good" or "bad" contractors were evaluated by QUEST for the September 1989 through December 1989 timeframe and four month average QUEST scores were computed. Also, the peer rating (A through F) was averaged to compute the equivalent of a Grade Point Average on a scale of 0.0 to 4.0 (4.0 equals straight A ratings for four consecutive months). The hypothesis that QUEST 3.0 was unable to distinguish, overall, between effective and ineffective contractors was overwhelmingly rejected. The level of significance for nonresident contractors was less than .0001 and for resident contractors the level of significance was .0039. In other words the probability of observing the degree of agreement between QUEST and the expert purely by chance is less than .0001 and .0039 respectively. QUEST 3.0 better matched expert opinion with the peer rating than the Total Scores. It appeared that the letter grades are more accurate in assessing performance than the numerical ratings. Figures 1 and 2 graphically summarize the test results. For example, Figure 1 shows that contractors identified as poor performers most frequently received a "D-" QUEST rating and good performers most frequently were rated as "A."

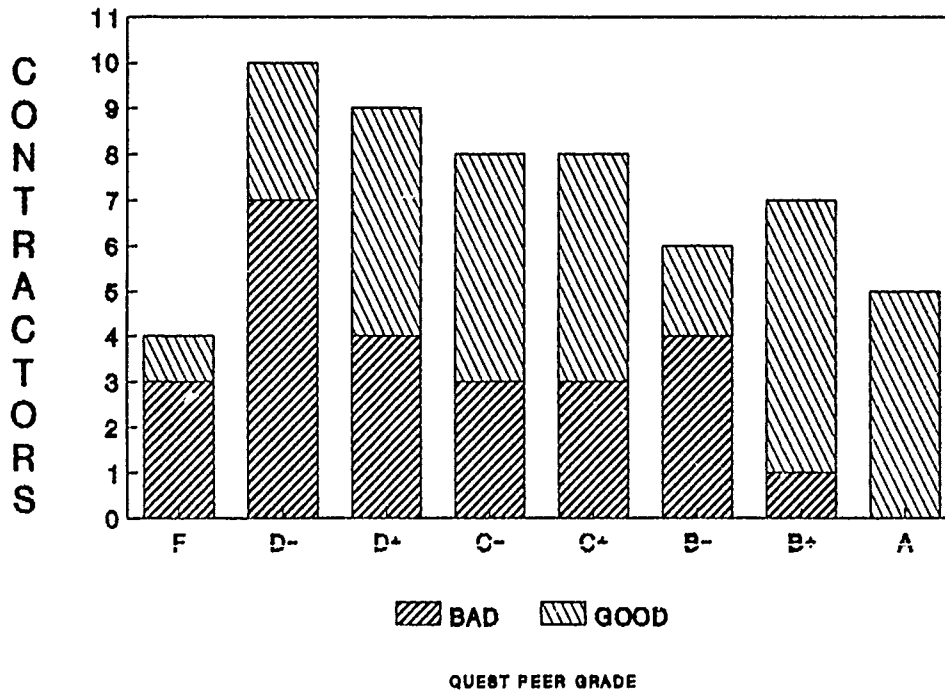
In an isolated number of cases, there is a disconnect between the QUEST rating and the expert's perception. A perceived "bad" contractor may have received an excellent QUEST rating and a perceived "good" contractor may be viewed unfavorably by QUEST. However, in most cases, the two evaluations agreed, causing sufficient correlation to pass statistical tests. Based on this test, prior experience with QUEST 2.0 and knowledge of the changes under QUEST 3.0, the SAG concluded that QUEST 3.0 was valid.

Figure 1

COMPARISON OF QUEST TO EXPERT
NONRESIDENT CONTRACTORS NOV89



COMPARISON OF QUEST TO EXPERT
RESIDENT CONTRACTORS NOV89



IV. CONCLUSIONS

Since QUEST 3.0 is a continuation of an established program, was designed with significant participation of experts, and has passed statistical tests of validity, it is concluded that:

- o QUEST 3.0 is implementable. It requires use of existing data only and has been run in a laboratory environment at DLA-DORO.

- o QUEST 3.0 is valid. Its results match the perception of experts in the field.

- o QUEST 3.0 adequately measures the QA effectiveness of DCMC contractors.

V. RECOMMENDATION. It is recommended that QUEST 3.0 be implemented throughout DCMC. It is also recommended that a feasibility study be conducted after IQUE is fully implemented to assess measures of in-house QA effectiveness or program effectiveness.

VI. BENEFITS

The incremental benefits of QUEST 3.0 from version 2.0 are based on the following logic. QUEST 3.0 has been selected as one of ten primary workload indicators in the proposed QA Resource Model currently under development. It is conservatively estimated that the use of this tool to justify and allocate quality resources could achieve comparable effectiveness with 1 percent fewer resources. Assuming that there are 7,000 QA specialists with an average grade of GS-10, step 5 with 29.55 percent fringe benefits, QUEST 3.0 benefits are:

$$\frac{1}{10} \times 1\% \times 7,000 \times \$32,098/\text{year} \times 1.2955$$

\$291,080 per year

In addition to these quantifiable benefits, QUEST 3.0 provides a more accurate measure of the contractor QA program. This information will be valuable to first line supervisors and upper management to evaluate the in-house IQUE program, to determine when and where remedial action is needed and to warn other acquisition elements where the IQUE program is not working. By measuring effectiveness based on the seven negative indicators (paragraph II.B.1-7), the QAR and the contractor will take actions to reduce the incidence of these negative indicators, leading to higher quality products and improved customer satisfaction.

VII. IMPLEMENTATION

An implementation plan was developed and approved by the SAC. It was proposed that DLA-DORO maintain the FORTRAN source code with the Job Control Language maintained by the DLA Systems Automation Center (DSAC). DSAC is responsible for exporting release 3.0 to the various field activities. Prior to release, the model will undergo environmental testing on actual field data at one site.

In September 1990, QUEST 3.0 was tested in a production environment on DCMC-West data. All aspects of the model were acceptable except the DMINS process. This part of the model must be installed by the Systems Branch, Program and Systems Management Division, Directorate of Quality Assurance (DLA-QRS) on-site and will be done in conjunction with site visits on other programs to conserve travel funds.

APPENDIX A
Study Advisory Group

<u>Name</u>	<u>Organization</u>	<u>Position</u>
Richard Zerilli	DLA-QRS	Chairman
Ronald DiPadova	DLA-QRS	HQ Staff
Halvor Elbeck	CHI-QU	HQ Staff
Jim Russell	DLA-LO	Study Director
Ted Tansey	DPSSO-C	Comptroller
Ray Butscher	DPSSO-C	Comptroller
Jerry Andrews	DCMR-ATL	Division Chief
Diane Stubblefield	DCMR-ATL	Staff
Steve Lapin	DCMR-BOS	Staff
Jack Rohan	DCMR-BOS	QAR
Sil Pontarelli	DCMR-CHI	Staff
Susan Nibling	DCMR-DAL	Staff
L.C. Long	DCMR-DAL	FLS
Kathy Rassmussen	DCMR-LA	Staff
Gary Achman	DCMR-LA	QAR
Bill Meuther	DCMR-NY	Staff
Chris Prendergast	DCMR-NY	FLS
Robert Lawson	DCMR-PHI	Staff
Brad Hillierman	DCMR-PHI	QAR
Terry Edwards	DCMR-STL	Staff

APPENDIX B

QUEST Source Code

<u>Section</u>	<u>Page</u>
PQDR Processor.....	B-2
File Merger	
JCL.....	B-10
FORTRAN.....	B-11
QUEST Generator	
JCL.....	B-14
Main Program.....	B-15
Subroutines	
Error Test.....	B-19
Peer Grouping.....	B-19
Problem Contractor.....	B-28
Initiatization.....	B-28
Preprocessor.....	B-30
TOPSIS.....	B-32
Time Link.....	B-34
Red Flags.....	B-35
Summary.....	B-38
Name of Contractor.....	B-40
Report Generator	
JCL.....	B-41
FORTRAN.....	B-42

```

//GDR6040B JOB (6040,GDR),'GROVER',CLASS=3,MSGCLASS=V
//*
//STEP1 EXEC FORTVCG,FVREGN=2500K,GOREGN=2000K,
//      PARM.GO='LET,NORES,EP=MAIN,SIZE=500000'
//FORT.SYSIN DD *
      CHARACTER*3 FIL1,M03
      CHARACTER*5 FSCM1
      CHARACTER*1 FIL2,FIL4,FIL5,FIL6,FRCN,FPRI,FAC,FDEF,PIINTP
      CHARACTER*12 FIL3
      CHARACTER*2 YR,FY
      CHARACTER*11 FIL7
      CHARACTER*4 FDAY51
      CHARACTER*26 FIL8
      CHARACTER*13 FIL9
1      FORMAT(A3,A5,A1,A1,A12,A1,A1,A2,A3,4A1,A11,A4,A26,A2,A1,A13)
2      READ(1,1,END=3) FIL1,FSCM1,FIL2,FRCN,FIL3,FPRI,
      +      FIL4,YR,M03,FIL5,FAC,FIL6,FDEF,
      +      FIL7,FDAY51,FIL8,FY,PIINTP,FIL9
      IF (FRCN.EQ.'S') THEN
      IF((PIINTP.EQ.'D').OR.(PIINTP.EQ.'G')) FY=YR
      IF((FY(1:1).LT.'O').OR.(FY(1:1).GT.'9')) FY=YR
      IF((FY(2:2).LT.'O').OR.(FY(2:2).GT.'9')) FY=YR
      WRITE(2,1) FIL1,FSCM1,FIL2,FRCN,FIL3,FPRI,
      +      FIL4,YR,M03,FIL5,FAC,FIL6,FDEF,
      +      FIL7,FDAY51,FIL8,FY,FIL9
      ENDIF
      GOTO 2
3      END
/*
//*
//GO.FTO1FOO1 DD DSN=GDR.GROVER.MDR.ATLCUM,MAY90,
//      DISP=SHR
//GO.FTO2FOO1 DD DSN=GDR.GROVER.ATL.MDRTEMP1,
//*      DISP=(NEW,CATLG,DELETE),
//      DISP=(,PASS),
//      DCB=(RECFM=FB,LRECL=90,BLKSIZE=18000),
//      UNIT=WORKD,SPACE=(CYL,(1,1),RLSE),
//      VOL=SER=WORKW1
//GO.FTO6FOO1 DD SYSOUT=*
//SYSOUT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
//*
//STEP2 EXEC PGM=IERRCOO
//SORTLIB DD DSN=SYS1.SORTLIB,DISP=SHR
//SYSUDUMP DD SYSOUT=*
//SORTMSG DD SYSOUT=*
//SYSOUT DD SYSOUT=*
//SORTIN DD DSN=GDR.GROVER.ATL.MDRTEMP1,
//      DISP=OLD
//SORTOUT DD DSN=GDR.GROVER.ATL.MDRTEMP2,
//      DISP=(,PASS),
//*      DISP=(NEW,CATLG,DELETE),
//      UNIT=WORKD,
//      DCB=(RECFM=FB,LRECL=90,BLKSIZE=18000),
//      SPACE=(TRK,(99,9),RLSE),
//      VOL=SER=WORKW1
//SORTWKO1 DD UNIT=WORKD,SPACE=(TRK,10)
//SORTWKO2 DD UNIT=WORKD,SPACE=(TRK,10)
//SORTWKO3 DD UNIT=WORKD,SPACE=(TRK,10)
//SYSIN DD *
      SORT FIELDS=(10,8,CH,A,19,3,CH,A,18,1,CH,A)
//*
//*
//STEP3 EXEC FORTVCG,FVREGN=2500K,GOREGN=2000K,
//      PARM.GO='LET,NORES,EP=MAIN,SIZE=500000'
//FORT.SYSIN DD *
C THERE IS A BUG IN THIS SECTION ON REOPENED MDRS THAT OCCURS IN
C REGIONS THAT HAVE MULTIPLE DODAACS ENTERING MDRS. THE SCR THAT
C CHANGED REOPENING MDRS ELIMINATES THE BUG. BUG ONLY APPLIES TO
C OLD DATA.
      CHARACTER*10 FIL1(2)
      CHARACTER*7 CNBRA(2)
      CHARACTER*1 CNBRB(2)
      CHARACTER*3 CNBRC(2)
      CHARACTER*1 FIL2(2)
      CHARACTER*1 PRI(2)
      CHARACTER*1 FIL3(2)
      CHARACTER*2 YR(2)

```

CHARACTER*3 MO3(2)	00750004
CHARACTER*1 FIL4(2)	00760004
CHARACTER*1 FAC(2)	00770004
CHARACTER*1 FIL5(2)	00780004
CHARACTER*1 FDEF(2)	00790004
CHARACTER*11 FIL6(2)	00800004
INTEGER FDAYS(2)	00810004
CHARACTER*16 FIL7A(2)	00820004
CHARACTER*10 FIL7B(2)	00830004
CHARACTER*2 FY(2)	00840004
CHARACTER*14 FIL8(2)	00850004
1 FORMAT(A10,A7,A1,A3,A1,A1,A1,A2,A3,A1,A1,A1,A1,	00860004
+A11,I4,A16,A10,A2,A14)	00870004
READ(1,1,END=5) FIL1(1),CNBRA(1),CNBRB(1),CNBRC(1),FIL2(1),	00880004
+PRI(1),FIL3(1),YR(1),MO3(1),FIL4(1),FAC(1),FIL5(1),FDEF(1),	00890004
+FIL6(1),FDAYS(1),FIL7A(1),FIL7B(1),FY(1),FIL8(1)	00900004
2 READ(1,1,END=3) FIL1(2),CNBRA(2),CNBRB(2),CNBRC(2),FIL2(2),	00910004
+PRI(2),FIL3(2),YR(2),MO3(2),FIL4(2),FAC(2),FIL5(2),FDEF(2),	00920004
+FIL6(2),FDAYS(2),FIL7A(2),FIL7B(2),FY(2),FIL8(2)	00930004
IF((CNBRA(1).EQ.CNBRA(2)).AND.(CNBRC(1).EQ.CNBRC(2))) THEN	00940004
FDAYS(1)=FDAYS(1)+FDAYS(2)	00950004
CNBRB(1)=CNBRB(2)	00960004
PRI(1)=PRI(2)	00970004
FAC(1)=FAC(2)	00980004
FDEF(1)=FDEF(2)	00990004
FY(1)=FY(2)	01000004
IF(FIL1(1).LE.' ') FIL1(1)=FIL1(2)	01010004
IF(FIL2(1).LE.' ') FIL2(1)=FIL2(2)	01020004
IF(FIL3(1).LE.' ') FIL3(1)=FIL3(2)	01030004
IF(FIL4(1).LE.' ') FIL4(1)=FIL4(2)	01040004
IF(FIL5(1).LE.' ') FIL5(1)=FIL5(2)	01050004
IF(FIL6(1).LE.' ') FIL6(1)=FIL6(2)	01060004
IF(FIL7A(1).LE.' ') FIL7A(1)=FIL7A(2)	01070004
IF(FIL7B(1).LE.' ') FIL7B(1)=FIL7B(2)	01080004
IF(FIL8(1).LE.' ') FIL8(1)=FIL8(2)	01090004
GOTO 2	01100004
ENDIF	01110004
WRITE(2,1) FIL1(1),CNBRA(1),CNBRB(1),CNBRC(1),FIL2(1),	01120004
+PRI(1),FIL3(1),YR(1),MO3(1),FIL4(1),FAC(1),FIL5(1),	01130004
+FDEF(1),FIL6(1),FDAYS(1),FIL7A(1),FIL7B(1),FY(1),FIL8(1)	01140004
FIL1(1)=FIL1(2)	01150004
CNBRA(1)=CNBRA(2)	01160004
CNBRB(1)=CNBRB(2)	01170004
CNBRC(1)=CNBRC(2)	01180004
FIL2(1)=FIL2(2)	01190004
PRI(1)=PRI(2)	01200004
FIL3(1)=FIL3(2)	01210004
YR(1)=YR(2)	01220004
MO3(1)=MO3(2)	01230004
FIL4(1)=FIL4(2)	01240004
FAC(1)=FAC(2)	01250004
FIL5(1)=FIL5(2)	01260004
FDEF(1)=FDEF(2)	01270004
FIL6(1)=FIL6(2)	01280004
FDAYS(1)=FDAYS(2)	01290004
FIL7A(1)=FIL7A(2)	01300004
FIL7B(1)=FIL7B(2)	01310004
FY(1)=FY(2)	01320004
FIL8(1)=FIL8(2)	01330004
GOTO 2	01340004
3 WRITE(2,1) FIL1(1),CNBRA(1),CNBRB(1),CNBRC(1),FIL2(1),	01350004
+PRI(1),FIL3(1),YR(1),MO3(1),FIL4(1),FAC(1),FIL5(1),	01360004
+FDEF(1),FIL6(1),FDAYS(1),FIL7A(1),FIL7B(1),FY(1),FIL8(1)	01370004
5 END	01380004
/*	01390004
/*	01400004
//GO.FTO1FOO1 DD DSN=GOR.GROVER.ATL.MDRTEMP2,	01410035
// DISP=OLD	01420005
//GO.FTO2FOO1 DD DSN=GOR.GROVER.ATL.MDRTEMP3,	01430035
// DISP=(NEW,CATLG,DELETE),	01440029
// DISP=(,PASS),	01450029
// DCB=(RECFM=FB,LRECL=90,BLKSIZE=18000),	01460004
// UNIT=WORKD,SPACE=(CYL,(1,1),RLSE),	01470022
// VOL=SER=WORKW1	01480022
//GO.FTO6FOO1 DD SYSOUT=*	01490004
//SYSOUT DD SYSOUT=*	01500004
//SYSUDUMP DD SYSOUT=*	01510004
//SYSPRINT DD SYSOUT=*	01520004
/*	01530004
/*	01540004

```

//STEP4 EXEC FORTVCG,FVREGN=2500K,GOREGN=2000K,                                01550004
//      PARM.GO='LET,NORES,EP=MAIN,SIZE=500000'                                01560004
//FORT.SYSIN DD *                                                                01570004
C *****                                                                    01580004
C      MODIFIED BY TLB ON 10/20/87                                              01590004
C                                                                              01600004
C                                                                              01610004
C      INPUT: MATERIAL DEFICIENCY REPORT (MDR) FILES FROM FIVE DCASRS          01620004
C      OUTPUT: LISTINGS OF NUMBER OF MDRS BY MANUFACTURERS BY MONTH            01630004
C                                                                              01640004
C *****                                                                    01650004
C      DOCUMENTATION SECTION                                                    01660004
C *****                                                                    01670004
C                                                                              01680004
C      THE PURPOSE OF THIS PROGRAM IS TO CREATE AN INPUT FILE FOR              01690004
C      ADDITIONAL PROCESSING. THE END RESULT OF THE                            01700004
C      PROGRAM IS A LISTING WHICH SHOWS HOW MANY MATERIAL                      01710004
C      DEFICIENCY REPORTS (MDR) ARE RECEIVED BY THE DCASR                     01720004
C      FOR A GIVEN MANUFACTURER IN A GIVEN MONTH. THE LENGTH                  01730004
C      OF TIME BETWEEN THE FISCAL YEAR OF THE CONTRACT AND THE                01740004
C      DATE THE MDR WAS RECEIVED IS TAKEN INTO CONSIDERATION,                 01750004
C      WITH THE SHORTER TIME CARRYING THE GREATER WEIGHT.                     01760004
C      IN ADDITION, THE NUMBER OF DAYS TAKEN TO CLOSE THE MDR                 01770004
C      ARE LISTED.                                                              01780004
C                                                                              01790004
C      THE LOGIC OF THE MAIN PROGRAM IS AS FOLLOWS:                           01800004
C                                                                              01810004
C      FIRST, IN THE SUBROUTINE CALLED "REVIEW", THE RAW DATA IS READ          01820004
C      FROM FILE ONE, SCREENED FOR ERRONEOUS DATA FIELDS                     01830004
C      AND THEN REWRITTEN TO FILE TWO.                                         01840004
C                                                                              01850004
C      SECOND, IN "READER", FILE TWO IS READ INTO AN ARRAY.                    01860004
C                                                                              01870004
C      THIRD, IN "JDATER", THE LAST THREE DIGITS OF THE JULIAN DATE            01880004
C      OF THE DATE MDR WAS RECEIVED IS CONVERTED INTO MONTHS                 01890004
C      ONE THROUGH TWELVE.                                                     01900004
C                                                                              01910004
C      FOURTH, IN "SORTER", THE MDRS ARE SORTED ACCORDING TO FSCM             01920004
C      AND THE YEAR AND MONTH THE MDRS WERE RECEIVED.                        01930004
C                                                                              01940004
C      FIFTH, IN "WEIGHT", THE TIME BETWEEN THE FISCAL YEAR OF THE             01950004
C      CONTRACT AND THE DATE THE MDR WAS RECEIVED IS                          01960004
C      COMPUTED TO REPRESENT THE AGE OF THE MDR. THE AGE                      01970004
C      IS USED TO WEIGHT THE MDR (SEE VARIABLE DICTIONARY FOR                 01980004
C      WEIGHTS USED)                                                            01990004
C                                                                              02000004
C      SIXTH, IN "WRITER", THE FSCM, MONTH AND YEAR THE MDR WAS RECEIVED,     02010004
C      THE WEIGHTED AVERAGE OF NUMBERS OF MDRS AND THE NUMBER OF             02020004
C      DAYS REQUIRED TO CLOSE THE MDR ARE WRITTEN TO FILE THREE.               02030004
C                                                                              02040004
C *****                                                                    02050004
C      VARIABLE DICTIONARY                                                      02060004
C *****                                                                    02070004
C                                                                              02080004
C      THE PROGRAM VARIABLES ARE DEFINED AS FOLLOWS:                           02090004
C                                                                              02100004
C      NAME      MEANING      REMARKS                                          02110004
C      ----      -
C      AC        ACTION CODE      ONE LETTER CODE                            02120004
C      DAYSCL    DAYS TO CLOSE    FOUR INTEGER NUMBER OF DAYS                02130004
C                                          TAKEN TO CLOSE OUT MDR              02140007
C                                          ONE LETTER CODE                      02150004
C      DEF       DEFECT CODE      02160004
C      FAC       ACTION CODE      02170004
C      FDAYS     DAYS TO CLOSE    02180007
C      FDAYS1    DAYS TO CLOSE    02190007
C      FDEF      DEFECT CODE      02200004
C      FIL1      FILLER #1        FILLERS WERE USED TO SIMPLIFY              02210004
C      FIL2      FILLER #2        RECORD MANIPULATIONS                      02220004
C      FIL3      FILLER #3        02230004
C      FIL4      FILLER #4        02240004
C      FIL5      FILLER #5        02250004
C      FIL6      FILLER #6        02260004
C      FIL7      FILLER #7        02270004
C      FIL8      FILLER #8        02280004
C      FIL9      FILLER #9        02290004
C      FPRI      PRIORITY CODE    02300004
C      FRCN      RECORD CONTROL NUMBER  FIRST LETTER IS 'S' IN ALL CASES 02310004
C      FSCM      MANUFACTURER CODE IDENTIFIES MANUFACTURER              02320004
C      FSCM1     MANUFACTURER CODE 02330004
C      FY        FISCAL YEAR      YEAR OF CONTRACT                        02340004

```

C	IEND	END SWITCH	SET TO '1' IN REEDER	02350004
C			TO PROPERLY END DATA READ IN	02360004
C	IFY	FISCAL YEAR		02370004
C	ITEMP1	TEMPORARY STORAGE	USED IN "SORTER"	02380004
C	ITEMP2	TEMPORARY STORAGE	USED IN "SORTER"	02390004
C	ITEMP3	TEMPORARY STORAGE	USED IN "SORTER"	02400004
C	ITEMP4	TEMPORARY STORAGE	USED IN "SORTER"	02410004
C	ITEMP5	TEMPORARY STORAGE	USED IN "SORTER"	02420004
C	ITEMP6	TEMPORARY STORAGE	USED IN "SORTER"	02430004
C	JJ	ARRAY PARAMETER	USED IN "WEIGHT" TO SET ARRAY	02440004
C			PARAMETERS IN AGER	02450004
C	JM	DO LOOP PARAMETER	USED IN "SORTER"	02460004
C	KYR	YEAR MDR RECEIVED		02470004
C	MDRCTR	MDR COUNTER	THE NUMBER OF VALID MDR RECORDS	02480004
C	MDRNUM	NUMBER OF MDRS	THE NUMBER OF MDR RECORDS INPUT	02490004
C	MDRVAL	NUMBER OF MDRS		02500004
C	MO1	MONTH MDR RECEIVED	THREE DIGIT INTEGER	02510004
C	MO2	MONTH MDR RECEIVED	TWO DIGIT CONVERTED INTEGER	02520004
C	MO3	MONTH MDR RECEIVED	THREE CHARACTER VARIABLE USED	02530004
C			IN "REVIEW" TO SCREEN DATA	02540004
C	NN	DO LOOP PARAMETER	USED IN "SORTER"	02550004
C	PRI	PRIORITY CODE		02560004
C	RCN	RECORD CONTROL NUMBER		02570004
C	SORKEY	SORT KEY		02580004
C	WT	WEIGHT	ARBITRARY WEIGHT ASSIGNER IN	02590004
C			"WEIGHT" FOR AGE OF MDRS	02600004
C	WTDMDR	WEIGHTED NUMBER OF MDRS		02610004
C	YR	YEAR MDR RECEIVED	TWO CHARACTER VARIABLE USED IN	02620004
C			"REVIEW"	02630004
C	*****			02640004
C	INITIALIZATION SECTION			02650004
C	*****			02660004
C	PURPOSE: TO INITIALIZE, DECLARE AND DEFINE PROGRAM VARIABLES			02670004
C				02680004
C	CHARACTER SORKEY(5000)=10,FSCM(5000)*5			02690004
C	CHARACTER*1 RCN(5000),PRI(5000),DEF(5000),AC(5000)			02700004
C	INTEGER KYR(5000),MO1(5000),MO2(5000),IFY(5000),DAYSCL(5000)			02710004
C	REAL WTDMDR(5000)			02720004
C	COMMON DAYSCL,IFY,KYR,MO1,MO2,WTDMDR			02730004
C	COMMON FSCM,RCN,PRI,AC,DEF,SORKEY			02740004
C	MDRNUM = 10000			02750004
C				02760005
C	*****			02770004
C				02780004
C	MAIN PROGRAM			02790004
C	*****			02800004
C				02810004
C	CALL REVIEW(MDRNUM,MDRVAL)			02820004
C	CALL READER(MDRVAL)			02830004
C	CALL JDATE(MDRVAL)			02840004
C	CALL SORTER(MDRVAL)			02850004
C	CALL WEIGHT(MDRVAL)			02860004
C	CALL WRITER(MDRVAL)			02870004
C	STOP			02880004
C	END			02890004
C				02900004
C	*****			02910004
C	SUBROUTINES			02920004
C	*****			02930004
C	SUBROUTINE REVIEW(MDRNUM,MDRVAL)			02940004
C	*****			02950004
C				02960004
C	PURPOSE: TO REVIEW RECORD FOR ERRONEOUS DATA INPUT			02970004
C	IF THE DATA MEETS THESE CRITERIA THE RECORD WILL			02980004
C	WRITTEN TO FILE NUMBER 2			02990004
C				03000004
C	CHARACTER*5 FSCM1			03010004
C	CHARACTER*1 FRCN,FPRI,FAC,FDEF			03020004
C	CHARACTER*2 YR,FY			03030004
C	CHARACTER*3 MO3			03040004
C	CHARACTER*4 FDAYS1			03050004
C				03060004
C	MDRVAL=0			03070004
C	IEND=0			03080004
C	DO 20 I=1,MDRNUM			03090004
C	IF (IEND.EQ.1) GOTO 25			03100004
C	CALL REEDER(FSCM1,MO3,FY,YR,FDAYS1,FRCN,FPRI,FAC,FDEF,IEND)			03110004
C	**** THAT PRIORITY CODE DOES NOT = 5, OR ZERO			03121028
C	IF(((FPRI.NE.'5').AND.(FPRI.NE.'0')) .AND.			03130028
C	**** THAT PRIORITY CODE DOES NOT = 5, BLANK OR ZERO			03130128

```

C      IF(((FPRI.NE.'5') .AND. (FPRI.NE.' ') .AND. (FPRI.NE.'O')) .AND. 03131037
C      **** THAT ACTION CODE IS F OR Z 03140005
C      + ((FAC.EQ.'F') .OR. (FAC.EQ.'Z')) .AND. 03150037
C      **** THAT DEFECT CODE IS A THRU M OR X 03160034
C      + ((FDEF.LT.'N') .OR. (FDEF.EQ.'X')) .AND. 03170034
C      **** THAT CONTRACT YEAR IS BETWEEN FY75 AND FY99 03180034
C      + ((FY.GE.'75') .AND. (FY.LE.'99')) THEN 03190037
      MDRVAL=MDRVAL+1 03210004
      CALL WRYTER(FSCM1,MO3,FY,YR,FDAYS1,FRCN,FPRI,FAC,FDEF) 03220004
      ELSE 03230004
      GO TO 20 03240004
      ENDIF 03250004
20    CONTINUE 03260004
25    RETURN 03270004
      END 03280004
C 03290004
C***** 03300004
      SUBROUTINE READER(FSCM1,MO3,FY,YR,FDAYS1,FRCN,FPRI,FAC,FDEF,IEND) 03310004
C***** 03320004
C 03330004
C      PURPOSE: INPUTS A RECORD 03340004
C 03350004
      CHARACTER*3 FIL1,MO3 03360004
      CHARACTER*5 FSCM1 03370004
      CHARACTER*1 FIL2,FIL4,FIL5,FIL6,FRCN,FPRI,FAC,FDEF 03380004
      CHARACTER*12 FIL3 03390004
      CHARACTER*2 YR,FY 03400004
      CHARACTER*11 FIL7 03410004
      CHARACTER*4 FDAYS1 03420004
      CHARACTER*26 FIL8 03430004
      CHARACTER*14 FIL9 03440004
C 03450004
      READ(1,101,END=11) FIL1,FSCM1,FIL2,FRCN,FIL3,FPRI, 03460004
      + FIL4,YR,MO3,FIL5,FAC,FIL6,FDEF, 03470004
      + FIL7,FDAYS1,FIL8,FY,FIL9 03480004
C 03490004
101  FORMAT(A3,A5,A1,A1,A12,A1,A1,A2,A3,A1,A1,A1,A11,A4,A26,A2,A14) 03500004
      GO TO 12 03510004
11  IEND=1 03520004
12  RETURN 03530004
      END 03540004
C 03550004
C***** 03560004
      SUBROUTINE WRYTER(FSCM1,MO3,FY,YR,FDAYS1,FRCN,FPRI,FAC,FDEF) 03570004
C***** 03580004
C 03590004
C      PURPOSE: TO WRITE OUTPUT TO FILE NUMBER 2 03600004
C 03610004
      CHARACTER*5 FSCM1 03620004
      CHARACTER*2 YR,FY 03630004
      CHARACTER*1 FIL2,FIL4,FIL5,FIL6,FRCN,FPRI,FAC,FDEF 03640004
      CHARACTER*3 MO3 03650004
      CHARACTER*4 FDAYS1 03660004
      WRITE(2,102) FSCM1,FRCN,FPRI,YR,MO3,FAC,FDEF,FDAYS1, 03670004
      + FY,FSCM1,YR,MO3 03680004
102  FORMAT(A5,A1,A1,A2,A3,A1,A1,A4,A2,A5,A2,A3) 03690004
      RETURN 03700004
      END 03710004
C 03720004
C***** 03730004
      SUBROUTINE READER(MDRVAL) 03740004
C***** 03750004
C 03760004
C      PURPOSE: INPUTS A RECORD FROM FILE NUMBER 2 03770004
C 03780004
      CHARACTER SORKEY(5000)*10,FSCM(5000)*5 03790004
      CHARACTER*1 RCN(5000),PRI(5000),DEF(5000),AC(5000) 03800004
      INTEGER KYR(5000),MD1(5000),MD2(5000),IFY(5000),DAYSCL(5000) 03810004
      REAL WTDMDR(5000) 03820004
      COMMON DAYSCL,IFY,KYR,MD1,MD2,WTDMDR 03830004
      COMMON FSCM,RCN,PRI,AC,DEF,SORKEY 03840004
      REWIND 2 03850004
C 03860004
      DO 10 I=1,MDRVAL 03870004
100  READ(2,102) FSCM(I),RCN(I),PRI(I), 03880004
      + KYR(I),MD1(I),AC(I),DEF(I),DAYSCL(I), 03890004
      + IFY(I),SORKEY(I) 03900004
102  FORMAT(A5,A1,A1,I2,I3,A1,A1,I4,I2,A10) 03910004
10  CONTINUE 03920004
20  RETURN 03930004

```


C	END	03940004
C	*****	03950004
C	SUBROUTINE JDATE(MDRVAL)	03960004
C	*****	03970004
C	PURPOSE: TO CONVERT JULIAN DATES INTO MONTHS 1 - 12	03980004
C		03990004
C	CHARACTER SORKEY(5000)*10,FSCM(5000)*5	04000004
	CHARACTER*1 RCN(5000),PRI(5000),DEF(5000),AC(5000)	04010004
	INTEGER KYR(5000),MO1(5000),MO2(5000),IFY(5000),DAYSCL(5000)	04020004
	REAL WTDMDR(5000)	04030004
	COMMON DAYSCL,IFY,KYR,MO1,MO2,WTDMDR	04040004
	COMMON FSCM,RCN,PRI,AC,DEF,SORKEY	04050004
C	DO 45 J=1,MDRVAL	04060004
	1 IF((MO1(J).GE.001).AND.(MO1(J).LE.031)) THEN	04070004
	MO2(J) = 01	04080004
	ELSE IF((MO1(J).GE.032).AND.(MO1(J).LE.059)) THEN	04090004
	MO2(J) = 02	04100037
	ELSE IF((MO1(J).GE.060).AND.(MO1(J).LE.090)) THEN	04110004
	MO2(J) = 03	04120037
	ELSE IF((MO1(J).GE.091).AND.(MO1(J).LE.120)) THEN	04130004
	MO2(J) = 04	04140037
	ELSE IF((MO1(J).GE.121).AND.(MO1(J).LE.151)) THEN	04150004
	MO2(J) = 05	04160037
	ELSE IF((MO1(J).GE.152).AND.(MO1(J).LE.181)) THEN	04170004
	MO2(J) = 06	04180037
	ELSE IF((MO1(J).GE.182).AND.(MO1(J).LE.212)) THEN	04190004
	MO2(J) = 07	04200037
	ELSE IF((MO1(J).GE.213).AND.(MO1(J).LE.243)) THEN	04210004
	MO2(J) = 08	04220037
	ELSE IF((MO1(J).GE.244).AND.(MO1(J).LE.273)) THEN	04230004
	MO2(J) = 09	04240037
	ELSE IF((MO1(J).GE.274).AND.(MO1(J).LE.304)) THEN	04250004
	MO2(J) = 10	04260037
	ELSE IF((MO1(J).GE.305).AND.(MO1(J).LE.334)) THEN	04270004
	MO2(J) = 11	04280037
	ELSE IF((MO1(J).GE.335).AND.(MO1(J).LE.366)) THEN	04290004
	MO2(J) = 12	04300037
	ENDIF	04310004
45	CONTINUE	04320037
50	RETURN	04330004
	END	04340004
C		04350004
C		04360004
C	*****	04370004
C	SUBROUTINE SORTER(MDRVAL)	04380004
C	*****	04390004
C	PURPOSE: TO PREPARE FOR LATER MERGING SORTING IS DONE	04400004
C	BY FSCM BY YEAR MDR RECEIVED AND BY MONTH	04410004
C		04420004
C	CHARACTER*10 ITEMP6	04430004
	CHARACTER*5 ITEMP1	04440004
	CHARACTER SORKEY(5000)*10,FSCM(5000)*5	04450004
	CHARACTER*1 RCN(5000),PRI(5000),DEF(5000),AC(5000)	04460004
	INTEGER KYR(5000),MO1(5000),MO2(5000),IFY(5000),DAYSCL(5000)	04470004
	REAL WTDMDR(5000)	04480004
	COMMON DAYSCL,IFY,KYR,MO1,MO2,WTDMDR	04490004
	COMMON FSCM,RCN,PRI,AC,DEF,SORKEY	04500004
	NN=MDRVAL-1	04510004
	DO 110 J=1,NN	04520004
	JM = MDRVAL-1	04530004
C	DO 120 I=1,JM	04540004
	IF (SORKEY(I).LE.SORKEY(I+1)) THEN	04550004
	GO TO 120	04560004
	ELSE	04570004
	ITEMP1 = FSCM(I)	04580004
	ITEMP2 = KYR(I)	04590004
	ITEMP3 = MO2(I)	04600004
	ITEMP4 = DAYSCL(I)	04610004
	ITEMP5 = IFY(I)	04620004
	ITEMP6 = SORKEY(I)	04630004
	FSCM(I) = FSCM(I+1)	04640004
	KYR(I) = KYR(I+1)	04650004
	MO2(I) = MO2(I+1)	04660004
	DAYSCL(I) = DAYSCL(I+1)	04670004
	IFY(I) = IFY(I+1)	04680004
		04690004
		04700004
		04710004
		04720004
		04730004

SORKEY(I) = SORKEY(I+1)	04740004
FSCM(I+1) = ITEMP1	04750004
KYR(I+1) = ITEMP2	04760004
MO2(I+1) = ITEMP3	04770004
DAYSCL(I+1) = ITEMP4	04780004
IFY(I+1) = ITEMP5	04790004
SORKEY(I+1) = ITEMP6	04800004
ENDIF	04810004
120 CONTINUE	04820004
110 CONTINUE	04830004
RETURN	04840004
END	04850004
C	04860004
C	04870004
.....	04880004
SUBROUTINE WEIGHT(MDRVAL)	04890004
C	04900004
C	04910004
C	04920004
C	04930004
PURPOSE: TO COMPUTE A WEIGHTED AVERAGE OF MDRS BY MANUFACTURER	04940004
BY MONTH MDR WAS RECEIVED	04950004
C	04960004
INTEGER MDRCTR(5000)	04970004
CHARACTER SORKEY(5000)*10,FSCM(5000)*5	04980004
CHARACTER*1 RCN(5000),PRI(5000),DEF(5000),AC(5000)	04990004
INTEGER KYR(5000),MO1(5000),MO2(5000),IFY(5000),DAYSCL(5000)	05000004
REAL WTDMDR(5000)	05010004
COMMON DAYSCL,IFY,KYR,MO1,MO2,WTDMDR	05020004
COMMON FSCM,RCN,PRI,AC,DEF,SORKEY	05030004
M=0	05040004
25 M = M + 1	05050004
IF (M.EQ.MDRVAL+1) GO TO 55	05060004
WTDMDR(M)=0.0	05070004
CALL SAMER(MDRVAL,MDRCTR,M)	05080004
L=MDRCTR(M)	05090004
DO 45 N=1,L	05100004
JJ=M	05110004
IF (L.EQ. 1) GOTO 30	05120004
JJ = M + N - 1	05130004
IF (N.EQ. 1) GOTO 30	05140004
DAYSCL(M) = DAYSCL(M) + DAYSCL(JJ)	05150004
30 CALL AGER(WT,JJ)	05160004
WTDMDR(M) = WTDMDR(M)+WT	05170004
45 CONTINUE	05180004
DAYSCL(M) = DAYSCL(M)/MDRCTR(M)	05190004
M=M+MDRCTR(M)-1	05200004
GO TO 25	05210004
55 RETURN	05220004
END	05230004
C	05240004
C	05250004
.....	05260004
SUBROUTINE SAMER(MDRVAL,MDRCTR,M)	05270004
C	05280004
C	05290004
C	05300004
C	05310004
C	05320004
PURPOSE: TO CHECK FOR RECORDS WITH THE SAME FSCM,	05330004
YEAR AND MONTH AND INCREMENT A COUNTER	05340004
CALLED MDRCTR TO TRACK THE NUMBER OF RECORDS	05350004
WITH THE SAME PARAMETERS. THIS TOTAL WILL	05360004
BE USED IN WAITER TO COMPUTE WEIGHTED AVERAGES	05370004
C	05380004
INTEGER MDRCTR(5000)	05390004
CHARACTER SORKEY(5000)*10,FSCM(5000)*5	05400004
CHARACTER*1 RCN(5000),PRI(5000),DEF(5000),AC(5000)	05410004
INTEGER KYR(5000),MO1(5000),MO2(5000),IFY(5000),DAYSCL(5000)	05420004
REAL WTDMDR(5000)	05430004
COMMON DAYSCL,IFY,KYR,MO1,MO2,WTDMDR	05440004
COMMON FSCM,RCN,PRI,AC,DEF,SORKEY	05450004
C	05460004
C	05470004
C	05480004
C	05490004
C	05500004
THIS DO LOOP IS LIMITED TO THE NUMBER	05510004
OF MDRS OF A GIVEN FSCM IN A MONTH	05520004
C	05530004
C	05540004
C	05550004
C	05560004
C	05570004
C	05580004
C	05590004
C	05600004
MDRCTR(M)=1	05610004
DO 60 K=1,100	05620004
IF ((FSCM(M).EQ.FSCM(M+K)).AND.(KYR(M).EQ.KYR(M+K)).AND.	05630004
+ (MO2(M).EQ.MO2(M+K))) THEN	05640004
MDRCTR(M) = MDRCTR(M) + 1	05650004
ELSE	05660004
GO TO 40	05670004
ENDIF	05680004
60 CONTINUE	05690004

```

40  RETURN
END
C
C*****
SUBROUTINE AGER(WT,JJ)
C*****
C
C  PURPOSE:  TO COMPUTE AN AGE AND CORRELATE IT TO A RESPECTIVE
C            WEIGHT
C
C      INTEGER AGE
C      CHARACTER SORKEY(5000)*10,FSCM(5000)*5
C      CHARACTER*1 RCN(5000),PRI(5000),DEF(5000),AC(5000)
C      INTEGER KYR(5000),MO1(5000),MO2(5000),IFY(5000),DAYSCL(5000)
C      REAL WTDMDR(5000)
C      COMMON DAYSCL,IFY,KYR,MO1,MO2,WTDMDR
C      COMMON FSCM,RCN,PRI,AC,DEF,SORKEY
C      WT=0
C      AGE=KYR(JJ)-IFY(JJ)+1
C      IF ((AGE.EQ.1).OR.(AGE.EQ.2)) WT = 1.0
C      IF (AGE.EQ.3) WT = .82
C      IF (AGE.EQ.4) WT = .47
C      IF (AGE.EQ.5) WT = .18
55  RETURN
END
C
C*****
SUBROUTINE WRITER(MDRVAL)
C*****
C
C  PURPOSE:  TO WRITE OUTPUT TO FILE NUMBER 3
C
C
C      CHARACTER SORKEY(5000)*10,FSCM(5000)*5
C      CHARACTER*1 RCN(5000),PRI(5000),DEF(5000),AC(5000)
C      INTEGER KYR(5000),MO1(5000),MO2(5000),IFY(5000),DAYSCL(5000)
C      REAL WTDMDR(5000)
C      COMMON DAYSCL,IFY,KYR,MO1,MO2,WTDMDR
C      COMMON FSCM,RCN,PRI,AC,DEF,SORKEY
C
C      WRITE THE FIRST RECORD
C      WRITE(3,190) FSCM(1),MO2(1),KYR(1),WTDMDR(1),DAYSCL(1)
C
C      WRITER THE REST OF THE FILE
C      DO 192 I=2,MDRVAL
C      IF ((FSCM(I).EQ.FSCM(I-1)).AND.(KYR(I).EQ.KYR(I-1))
C      + .AND.(MO2(I).EQ.MO2(I-1))) THEN
C      GO TO 192
C      ELSE
C      WRITE(3,190) FSCM(I),MO2(I),KYR(I),WTDMDR(I),DAYSCL(I)
C      ENDIF
190  FORMAT(A5,2X,I2,2X,I3,2X,F5.2,2X,I4)
192  CONTINUE
RETURN
END
/*
//GO.FTO1FOO1 DD DSN=GDR.GROVER.ATL.MDRTEMP3,DISP=OLD
//GO.FTO2FOO1 DD DSN=GDR.GROVER.ATI.MDRTEMP4,
//          DISP=(,DISP),
//          DISP=(NEW,CATLG,DELETE),
//          UNIT=WORKD,SPACE=(CYL,(1,10),RLSE),
//          DCB=(RECFM=FB,LRECL=30,BLKSIZE=3000),
//          VOL=SER=WORKW1
//GO.FTO3FOO1 DD DSN=GDR.GROVER.MDR.INATL.MAY90,
//          DISP=(NEW,CATLG,DELETE),
//          UNIT=WORKD,SPACE=(TRK,(1,1),RLSE),
//          VOL=SER=DOROC2,
//          DCB=(RECFM=FB,LRECL=27,BLKSIZE=2700)
//GO.FTO6FOO1 DD SYSOUT=*
//SYSOUT DD SYSOUT=*
//SYSDUMP DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
/*
//

```

//GOR6040L JOB (6040,GOR),'GROVER',CLASS=1,MSGCLASS=V	00010060
//RUNFTN EXEC FORTVCG	00030005
//FORT.SYSIN DD DSN=GOR.GROVER.QUEST3(MERGPROD),DISP=SHR	00040061
//GO.FTO1FOO1 DD DSN=GOR.GROVER.NPP1.LA.SORT,DISP=SI R	00044060
//GO.FTO2FOO1 DD DSN=GOR.GROVER.FAC.LA.MAR90,DISP=SH.1	00045062
//GO.FTO3FOO1 DD DSN=GOR.GROVER.MDR.INLA.MAY90,DISP=SHR	00046060
//GO.FTO4FOO1 DD DSN=GOR.GROVER.LA.INQUEST.MAY90.	00047060
// DISP=(NEW,CATLG,DELETE),DCB=(LRECL=388,RECFM=FB,BLKSIZE=3880),	00048060
// UNIT=WORKD,SPACE=(CYL,(5,10),RLSE),VOL=SER=WORKW1	00048160
//* UNIT=TAPE,LABEL=EXPDT=93001	00049060
//GO.FTO6FOO1 DD SYSOUT=*	00049129
//SYSOUT DD SYSOUT=*	00049229
//SYSUDUMP DD SYSOUT=*	00049329
//SYSPRINT DD SYSOUT=*	00049429
//	00049529

C	THIS MODULE MERGES THE FACILTY PROFILE, THE HISTORY FILE AND	00010033
C	THE MDR FILE BY FSCM TO PRODUCE A MASTER DATA FILE FOR THE MODEL.	00020033
C		00030033
C	THIS IS A FILE MERGER USED FOR QUEST III.MEMBER FOR(MATCH7)	00040051
C	THIS IS THE PRODUCTION VERSION TO BE USED WITH THE NEW QAMIS.	00041052
C	SPECIAL PROCESSES ARE PICKED UP ON FACILITY PROFILE.	00050048
C	ALSO, GOVT VERIFICATION CODE IS PICKED UP TO IDENTIFY IQUE FACILITIES.	00051049
C		00052049
C	DECLARE AND ARRAY VARIABLES	00060033
C	ARRAYS ARE DIMENSIONED TO ACCEPT NO MORE THAN TEN YEARS OF DATA	00070033
C		00080033
C	A AND AA AND AAA ARE FSCM OR CAGE	00090046
C	B AND BB ARE YEAR	00100044
C	C AND CC ARE MONTH	00110044
C		00120044
	CHARACTER A*5,AA(120)*5,DUM1(120)*10,DUM2(120)*4,GVN(120)*4	00130049
	CHARACTER DUM3(120)*341,AOPER(120)*2,OPER*2,FLUF1*10	00140052
	CHARACTER FLUF2*16,FLUF3*12,FLUF4*111,AAA*5,SPEC*4,GV*4,FLUF5*9	00150049
	INTEGER BB(120),CC(120),B,C,E,NQAR(120),DAYSCL(120),ISPEC(120)	00160048
	REAL WMDR(120)	00170037
C		00180010
C	*** INITIALIZE AND SET DEFAULT VALUES ***	00190010
C		00200033
	IFSCM=0	00210010
	IYR=0	00220010
	IMONTH=0	00230010
C	NEXT THREE VARIABLES ARE USED TO SUPPRESS FILE READ	00240033
	ICTL=1	00250012
	JCTL=1	00260021
	KCTL=1	00270021
	AAA='AAAAA'	00280031
	A='AAAAA'	00290041
	ICOUNT=0	00300020
	ISKIP=1	00310016
	DO 5 II=1,120	00320016
	AA(II)='00000'	00330016
	BB(II)=0	00340016
	CC(II)=0	00350016
	5 CONTINUE	00360016
C	SET DEFAULT VALUES	00370034
	10 DO 20 I=1,120	00380010
	NQAR(I)=0	00390012
	ISPEC(I)=0	00391048
	GVN(I)=' '	00392049
	WMDR(I)=0.0	00400035
	DAYSCL(I)=0	00410035
	AOPER(I)='AB'	00420012
	20 CONTINUE	00430010
	INPP1=1	00440010
C		00450033
C	*** INITIALIZATION COMPLETE ***	00460010
C		00470010
C	*** STEP 2 ***	00480010
C	*** READ NPP1 FILE RECORD TO BEGIN MATCHING PROCESS ***	00490010
C		00500033
C	THE HISTORY FILE IS SKIPPED IF THE FIRST HISTORY RECORD FOR A FSCM	00510033
C	HAS ALREADY BEEN READ.SEE 'RESET COUNTERS AND DEFAULT' SECTION	00520033
	100 IF(ISKIP-1) 109,101,101	00530016
	101 READ(1,102,END=330) DUM1(INPP1),AA(INPP1),DUM2(INPP1),BB(INPP1)	00540046
	1,CC(INPP1),DUM3(INPP1)	00550046
	102 FORMAT(A10,A5,A4,I2,I2,A341)	00560052
	ISKIP=1	00570016
C	TIMING CONVENTION FOR HISTORY RECORDS IS JNPP1 IS THE CURRENT RECORD	00580033
C	AND KNPP1 IS THE PREVIOUS RECORD, INPP1 IS THE NEXT RECORD TO READ.	00590033
	JNPP1=INPP1	00600010
	KNPP1=INPP1-1	00610010
	INPP1=INPP1+1	00620010
C	UPON IDENTIFICATION OF A FIRST FSCM RECORD, ATTEMPT TO MERGE WITH	00630033
C	FACILTY PROFILE	00640033
	IF(JNPP1 .EQ. 1) GO TO 110	00650021
C	IF CURRENT AND PREVIOUS RECORDS HAVE DIFFERENT FSCMS, WRITE PREVIOUS	00660033
C	FSCM RECORDS TO TAPE,IF THE SAME ATTEMPT TO MERGE WITH MDR FILE.	00670033
	IF(AA(JNPP1) .GT. AA(KNPP1)) GOTO 190	00680022
	IF(AA(JNPP1) .EQ. AA(KNPP1)) THEN	00690012
	IF((AA(JNPP1) .GT. A) .OR. ((AA(JNPP1) .EQ. A) .AND. ((BB(JNPP1)	00700047
1	.GT. B) .OR. ((CC(JNPP1) .GT. C) .AND. (BB(JNPP1) .EQ. B))))	00710047
2	GOTO 104	00720040
	IF((AA(JNPP1) .LT. A) .OR. ((AA(JNPP1) .EQ. A) .AND. ((BB(JNPP1)	00730047
1	.LT. B) .OR. ((CC(JNPP1) .LT. C) .AND. (BB(JNPP1) .EQ. B))))	00740047

2	GOTO 101	00750040
	GO TO 240	00760040
	ELSE	00770012
	GO TO 310	00780012
	ENDIF	00790012
104	ICTL=1	00800021
	GO TO 200	00810021
C		00820033
C	*** STEP 3 ***	00830033
C	*** READ FACILITY PROFILE FILE ***	00840033
C		00850033
109	INPP1=2	00860016
C	IF END OF PROFILE HAS BEEN REACHED OR HISTORY LAGS PREVIOUS PROFILE	00870033
C	SKIP THIS SECTION	00880033
110	IF(JCTL .EQ. 0) GOTO 200	00890021
	IF(AAA .GT. AA(JNPP1)) GOTO 200	00900021
C	IF HISTORY MATCHES PREVIOUS PROFILE, SKIP THE READ AND MERGE RECORDS	00910033
	IF(AAA .EQ. AA(JNPP1)) GOTO 130	00920021
C	READ A NEW PROFILE RECORD. IF FSCM MATCHES HISTORY FSCM, MERGE	00930033
C	RECORDS. OTHERWISE REPEAT OR GO LOOK AT MDR FILE	00940033
	READ(2,120,END=180) FLUF1,AAA,FLUF2,MQAR,OPER,FLUF3,SPEC,FLUF4,	00950049
	1GV,FLUF5	00951049
120	FORMAT(A10,A5,A16,I2,A2,A12,A4,A111,A4,A9)	00960049
C	IF NO MATCH IS FOUND, DEFAULTS VALUES ARE KEPT. SEE INITIALIZATION.	00970033
	IF (AAA .GT. AA(JNPP1)) GOTO 200	00980021
	IF(AAA .EQ. AA(JNPP1)) THEN	00990012
	GO TO 130	01000012
	ELSE	01010012
	GO TO 110	01020012
	ENDIF	01030012
C	A MATCH HAS BEEN MADE. APPEND PROFILE VARIABLES TO HISTORY RECORD.	01040033
130	DO 140 J=1,120	01050010
	NQAR(J)=MQAR	01060012
	ACPER(J)=OPER	01070012
	IF(SPEC .NE. ' ') ISPEC(J)=1	01071048
	GVN(J)=GV	01072049
140	CONTINUE	01080010
	GO TO 200	01090028
C	ONCE END OF PROFILE HAS BEEN REACHED, JCTL TURNS OFF FUTURE READS	01100033
180	JCTL=0	01110030
	GO TO 200	01120023
C		01130033
C	*** STEP 4 ***	01140033
C	*** WRITE RECORD TO TAPE**	01150033
C		01160033
190	DO 192 L=1,KNPP1	01170010
	WRITE(4,191) DUM1(L),AA(L),DUM2(L),BB(L),CC(L),DUM3(L),NQAR(L)	01180012
1	,ACPER(L),ISPEC(L),GVN(L),WMDR(L),DAYSCL(L),L,KNPP1	01190049
191	FORMAT(A10,A5,A4,I2,I2,A341,I2,A2,I1,A4,F5.2,I4,2I3)	01200052
	ICOUNT=ICOUNT+1	01210016
	AA(L)='00000'	01220016
	BB(L)=0	01230016
	CC(L)=0	01240016
192	CONTINUE	01250010
C	*** RESET COUNTERS AND DEFAULTS ***	01260010
	AA(1)=AA(JNPP1)	01270016
	BB(1)=BB(JNPP1)	01280016
	CC(1)=CC(JNPP1)	01290016
	DUM1(1)=DUM1(JNPP1)	01300016
	DUM2(1)=DUM2(JNPP1)	01310016
	DUM3(1)=DUM3(JNPP1)	01320016
	JNPP1=1	01330010
	ISKIP=0	01340016
	GO TO 10	01350010
C	*** END OF STEP 4 ***	01360033
C		01370010
C	*** MATCH TO MDR FILE BY FSCM AND MONTH AND YEAR ***	01380033
C	*** STEP 5 ***	01390033
C		01400033
C	READ IS SUPPRESSED IF THE END OF MDR FILE HAS BEEN REACHED OR	01410033
C	HISTORY FILE LAGS MDR FILE	01420033
200	IF(ICTL-1) 215,201,201	01430012
201	IF(KCTL .EQ. 0) GOTO 101	01440021
C	READ MDR RECORD. PICK UP FSCM, MONTH, YEAR, MDR COUNT AND DAYS	01450046
	READ(3,210,ERR=296,END=295) A,C,B,D,E	01460046
210	FORMAT(A5,2X,I2,2X,I3,2X,F5.2,2X,I4)	01470035
215	IF(A .GT. AA(JNPP1)) GOTO 270	01480016
	IF(A .EQ. AA(JNPP1)) THEN	01490016
	GO TO 220	01500012
	ELSE	01510012

ICTL=1	01520038
GO TO 201	01530016
ENDIF	01540012
220 IF(B-BB(JNPP1)) 201,230,280	01550046
230 IF(C-CC(JNPP1)) 201,240,290	01560046
C A MATCH HAS BEEN MADE. APPEND MDR COUNT TO HISTORY RECORD	01570033
240 WMDR(JNPP1)=D	01580016
DAYSCL(JNPP1)=E	01590035
ICTL=1	01600012
GO TO 101	01610016
C THE HISTORY FILE LAGS THE MDR FILE. GO BACK AND READ ANOTHER	01620033
C HISTORY RECORD.MDR COUNT OF CURRENT HISTORY DEFAULTS TO ZERO	01630033
270 IFSCM=IFSCM+1	01640010
ICTL=0	01650012
GO TO 101	01660016
280 IYR=IYR+1	01670010
ICTL=0	01680021
GO TO 101	01690016
290 IMONTH=IMONTH+1	01700010
ICTL=0	01710021
GO TO 101	01720016
295 KCTL=0	01730023
GO TO 101	01740021
296 GO TO 200	01750025
C *** END OF STEP 5 ***	01760033
C	01770010
C *** END PROGRAM ***	01780010
310 WRITE(6,320) AA(KNPP1),AA(JNPP1)	01790019
320 FORMAT(5X,'NPP1 FILE NOT SORTED BY FSCM IN ASCENDING ORDER ',A5,	01800010
15X,A5)	01810010
GO TO 190	01820036
330 KNPP1=JNPP1	01830021
DO 331 N=1,JNPP1	01840021
WRITE(4,332) DUM1(N),AA(N),DUM2(N),BB(N),CC(N),DUM3(N),NOAR(N)	01850012
1 ,AOPER(N),ISPEC(N),GVN(N),WMDR(N),DAYSCL(N),N,KNPP1	01860049
332 FORMAT(A10,A5,A4,I2,I2,A34,I2,A2,I1,A4,F5.2,I4,2I3)	01870052
ICOUNT=ICOUNT+1	01880016
331 CONTINUE	01890011
WRITE(6,333) IFSCM,IYR,IMONTH,ICOUNT	01900016
333 FORMAT(5X,4I10)	01910016
335 STOP	01920017
END	01930000

```

//GOR6040S JOB (6040,GOR),'GROVER',CLASS=1,MSGCLASS=V
//RUNFTN EXEC FORTVCG,FVREGN=2500K,GOREGN=2000K
// PARM.GD='LET,NORES,EP=MAIN,SIZE=500000'
//FORT.SYSIN DD DSN=GOR.GROVER.QUEST3(Q3PROD),DISP=SHR
//GO.FTO1FOO1 DD DSN=GOR.GROVER.LA.INQUEST.MAY90,DISP=SHR
//GO.FTO2FOO1 DD DSN=GOR.GROVER.LA.MAY90.DISKZX,UNIT=WORKD,
// DISP=(NEW,CATLG,DELETE),
// * DISP=(NEW,CATLG,DELETE),VOL=SER=WORKW2,
// DCB=(LRECL=235,RECFM=FB,BLKSIZE=2350),
// SPACE=(CYL,(19,9),RLSE)
//GO.FTO3FOO1 DD DSN=GOR.GROVER.QUEST3(PGMCTL),DISP=SHR
//GO.FTO4FOO1 DD DSN=GOR.GROVER.LALBL.ZX,UNIT=WORKD,
// DISP=(NEW,CATLG,DELETE),
// * DISP=(NEW,CATLG,DELETE),VOL=SER=WORKW2,
// DCB=(LRECL=236,RECFM=FB,BLKSIZE=2360),
// SPACE=(CYL,(9,9),RLSE)
//GO.FTO9FOO1 DD DSN=GOR.GROVER.LA.TOPZX,UNIT=WORKD,
// DISP=(NEW,CATLG,DELETE),
// * DISP=(NEW,CATLG,DELETE),VOL=SER=WORKW2,
// DCB=(LRECL=172,RECFM=FB,BLKSIZE=17200),
// SPACE=(CYL,(3,1),RLSE)
//GO.FT11FOO1 DD DSN=GOR.GROVER.LA.OUTZX,UNIT=WORKD,
// DISP=(NEW,CATLG,DELETE),
// * DISP=(NEW,CATLG,DELETE),VOL=SER=WORKW2,
// DCB=(LRECL=156,RECFM=FB,BLKSIZE=15600),
// SPACE=(CYL,(2,1),RLSE)
//GO.FTO8FOO1 DD DSN=GOR.GROVER.ALERT.DMAR90,DISP=SHR
//GO.FT10FOO1 DD DSN=GOR.GROVER.MASTER.JUL90,DISP=SHR
//GO.FTO6FOO1 DD SYSOUT=*
//SYSOUT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
// *
//STEP2 EXEC FORTVCG,FVREGN=2500K,GOREGN=2000K,
// PARM.GD='LET,NORES,EP=MAIN,SIZE=500000'
//FORT.SYSIN DD DSN=GOR.GROVER.QUEST3(ADDNAME),DISP=SHR
//GO.FTO1FOO1 DD DSN=GOR.GROVER.LA.OUTZX,DISP=SHR
//GO.FTO2FOO1 DD DSN=GOR.GROVER.FSCM.ALL,DISP=SHR
//GO.FTO3FOO1 DD DSN=GOR.GROVER.LA.PRESORT.ZX,UNIT=WORKD,
// DISP=(NEW,CATLG,DELETE),
// DCB=(LRECL=178,RECFM=FB,BLKSIZE=17800),
// SPACE=(CYL,(3,1),RLSE)
// *
// * *****
// * THIS PROGRAM WILL SORT ON DATE AND QAORG CODE *
// * AND TYPE AND EFFECTIVENESS. *
// * *****
// *
// * *****
// * THIS SEGMENT SORTS GOR.GROVER.LA.PRESORT.ZX *
// * *****
// *
//STEP2 EXEC PGM=IERRCOO,PARM='MSG=AP',REGION=2000K
//SORTLIB DD DSN=SYS1.SORTLIB,DISP=SHR
//SYSOUT DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
//SORTIN DD DSN=GOR.GROVER.LA.PRESORT.ZX,DISP=SHR
// *
//SORTOUT DD DSN=GOR.GROVER.LA.REPZX,DISP=(NEW,CATLG,DELETE),
// SPACE=(CYL,(1,1),RLSE),UNIT=WORKD,VOL=SER=WORKW2
//SORTWK01 DD UNIT=WORKD,SPACE=(CYL,10)
//SORTWK02 DD UNIT=WORKD,SPACE=(CYL,10)
//SORTWK03 DD UNIT=WORKD,SPACE=(CYL,10)
//SORTWK04 DD UNIT=WORKD,SPACE=(CYL,10)
//SORTWK05 DD UNIT=WORKD,SPACE=(CYL,10)
//SYSIN DD *
SORT FIELDS=(20,3,CH,A,17,3,CH,A,1,3,CH,A,10,1,CH,D,110,7,CH,A)
END
//

```

```

00010099
00020099
00021099
00040099
00050099
00062099
00063099
00063199
00064099
00065099
00070099
00081099
00082099
00082199
00083099
00084099
00085099
00086099
00086199
00087099
00088099
00089099
00089199
00089299
00089399
00089499
00089599
00089699
0032C399
00330099
00340099
00350099
00360099
00360199
00360299
00361099
00362099
00363099
00364099
00365099
00366099
00367099
00370099
00380099
00390099
00400099
00410099
00420099
00430099
00440099
00450099
00460099
00470099
00480099
00490099
00500099
00510099
00520099
00530099
00540099
00550099
00560099
00570099
00580099
00590099
00600099
00610099
00620099
00630099
00640099

```



```

C      *** THIS PROGRAM IS THE FIELDIED VERSION FOR QUEST III          00010099
C      *** THIS PROGRAM USES THE DCMC DATA STRUCTURE, NOT THE DASC.    00020099
C      *** MEMBER Q3PROD IS THE FIELDIED VERSION.                      00030099
C                                                                           00040098
C HAS A FIVE CHARACTER QAR CODE                                         00050099
C                                                                           00060022
C      ***THIS MODULE IS RUN AFTER A MASTER TAPE HAS BEEN CREATED      00070002
C      BY MERGING THE MIS WITH THE MDR AND FACILITY PROFILE.***        00080002
C                                                                           00090002
C                                                                           00100002
C      DECLARE AND ARRAY VARIABLES                                       00110002
C                                                                           00120002
C      CHARACTER DCASR(120)*6,QAORG(120)*3,FSCM(120)*6,TYP(120)*1,    00130002
C      1CMDTY(120)*2,PVN(120)*1,ADCASR*6,BFSCM(2000)*5,OPER(120)*2,    00140002
C      2FLAG(120)*18,DLRIN(120)*12,DLROUT(120)*12,DLROH(120)*12,      00150082
C      3GVN(120)*4,QAR(120)*5                                           00160099
C      INTEGER MONTH(120),YEAR(120),PLANHR(120),LOTINS(120),PVIHR(120), 00170002
C      1PEHR(120),PRHR(120),AQDR(120),BQDR(120),CQDR(120),DQDR(120),    00180002
C      2EQDR(120),QDRHR(120),TVLHR(120),TNGHR(120),FMSHR(120),ADMNHR(120), 00190002
C      3SHIPMT(120),WD(120),WDHR(120),INTHR(120),REINHR(120),VISIT(120), 00200002
C      4MRBHR(120),ECP(120),ECPHR(120),MDRHR(120),MTGHR(120),MRB(120), 00210002
C      5POHR(120),PCO(120),CAO(120),CONTR(120),DEGREE(120),           00220099
C      6ACNTRT(120),BCNTRT(120),OCNTRT(120),ISTRAT(120),ONHAND(120),    00230026
C      7QALIIN(120),QALIRE(120),ADNHND(120),BONHND(120),ODNHND(120),    00240002
C      8NQDR(120),PVINP(120),PEELNP(120),NOAR(120),IPRNT(120),         00250099
C      9ISEQ(120),RECS(120),DAYSCL(120),STARTM,STARTY,ENDMO,ENDYR      00260002
C      INTEGER IPEER(120),ISPEC(120),AVLHR(120),PA(120),CAR(120)      00270099
C      REAL ST(120),EPA(120),DEVN(120),TOPCA(120),CARATE(120),         00280099
C      1WDRATE(120),ECPRAT(120),AMRBRA(120),EPARAT(120),DEVNRA(120),    00290099
C      2WMDRRA(120),TOPEPA(120),TOPMRB(120),TOPWD(120),TOPDEV(120),    00300099
C      3TOPECP(120),WMDR(120),TOPMDR(120),A(999,38),AIDEAL(4,7),      00310052
C      4AWORST(4,7),BIDEAL(4,7),BWORST(4,7),WEIGHT(7),TOPSCR(14,120)   00320099
C      COMMON /RVAR/A                                                    00330002
C      COMMON /CHT2/BFSCM,FLAG                                           00340002
C                                                                           00350002
C      ***THIS MODULE SELECTS RELEVANT DATA RECORDS FROM THE INPUT FILE, 00360002
C      PERFORMS EDIT CHECKS AND WRITES THE RECORD TO A DISK FILE.***    00370002
C                                                                           00380002
C      ***STEP 1 ***                                                    00390002
C      ***INITIALIZE PARAMETERS                                         00400002
C                                                                           00410002
C      STARTM=99                                                         00420002
C      STARTY=99                                                         00430002
C      ENDMO=99                                                         00440002
C      ENDYR=99                                                         00450002
C      MONTHS=12                                                         00460002
C      ***EACH DCASR CODE SHOULD BE INSERTED NEXT; CREATING A UNIQUE PGM 00470002
C      ***OMIT NEXT LINE FOR DLA-WIDE MODEL****                        00480002
C      ADCASR='S1102A'                                                  00490002
C      LCOUNT=0                                                         00500002
C      NCOUNT=0                                                         00510002
C      KCOUNT=0                                                         00520002
C      JERROR=0                                                         00530002
C                                                                           00540002
C      ***ENTER TIME FRAME LIMITS ***                                    00550002
C      AN EXTERNAL FILE MUST BE SET UP TO CONTAIN DATES TO CONTROL MODEL 00560002
C      STARTM AND STARTY ARE THE MONTH AND YEAR TO BEGIN MEASURING QA EFF 00570002
C      ENDMO AND ENDYR ARE THE MONTH AND YEAR THE MODEL STOPS          00580002
C      MONTHS ARE THE MINIMUM NUMBER OF DATA POINTS OF HISTORY NEEDED  00590002
C      CUTYR DELETES ALL DATA PRIOR TO CY SPECIFIED                   00600002
C      LONG INDICATES WHETHER OR NOT BACKUP FILES ARE CREATED.         00610002
C      ICIP SPECIFIES WHETHER OR NOT ALERT FILE IS USED.               00620002
C                                                                           00630002
C      READ(3,10) STARTM,STARTY                                         00640002
C      PERFORM EDIT CHECKS ON DATES ENTERED                             00650002
C      CALL TIMCHK(STARTM,STARTY,JERROR)                                00660002
C      IF(JERROR.EQ. 1) GOTO 30                                          00670002
C      READ(3,10) ENDMO,ENDYR                                           00680002
C      PERFORM EDIT CHECKS ON DATES ENTERED                             00690002
C      IF(STARTY.GT. ENDYR) GOTO 30                                      00700002
C      CALL TIMCHK(ENDMO,ENDYR,JERROR)                                  00710002
C      IF(JERROR.EQ.1) GOTO 30                                          00720002
C      READ(3,10) MONTHS,ICUTYR                                         00730002
C      IF(MONTHS.LT. 3) GOTO 20                                          00740002
C      IF(MONTHS.GT. 99) GOTO 20                                         00750002
C      READ(3,11) ICIP,LONG                                              00760016
C      WRITE(6,16) STARTM,STARTY,ENDMO,ENDYR                           00770002
C      GO TO 100                                                         00780002
C      10 FORMAT(2I2)                                                    00790002

```

```

11 FORMAT(2I1)
16 FORMAT(1X,'INPUT ACCEPTED NORMALLY, NO DEFAULTS. START',2I3,2X,
1'END',2I3)
20 WRITE(6,25) MONTHS
25 FORMAT(1X,'HISTORY OUTSIDE ALLOWABLE RANGE',I2)
GO TO 202
30 WRITE(6,35) STARTM,STARTY,ENDMO,ENDYR
35 FORMAT(1X,'DATES NOT ACCEPTED, TRY AGAIN',4I4)
GO TO 202

C
C *** STEP 2***
C ***READ A RECORD FROM MERGED DATA FILE PRODUCTION VERSION***
C ***CHECK FOR MISSING OR INVALID DATA ***
C ***DELETE RECORDS WITH NO HOURS REPORTED ***
C
100 DO 195 I=1,120
    IPRNT(I)=0
    READ(1,101,END=202) DCASR(I),QAORG(I),FSCM(I),TYP(I),
    1CMDTY(I),PVN(I),YEAR(I),MONTH(I),PLANHR(I),LOTINS(I),
    2PVIHR(I),PEHR(I),PRHR(I),AQDR(I),BQDR(I),CQDR(I),DQDR(I),
    3EQDR(I),QDRHR(I),TVLHR(I),TNGHR(I),FMSHR(I),ADMNHR(I),SHIPMT(I),
    4WD(I),WDHR(I),INTHR(I),REINHR(I),VISIT(I),MRBHR(I),ECP(I),ECPHR(I),
    5MDRHR(I),MTGHR(I),MRB(I),POHR(I),PCO(I),CAO(I),AVLHR(I),SI(I),
    6CONTR(I),DLRIN(I),DLROUT(I),ACNTRT(I),BCNTRT(I),OCNTRT(I),QALIIN(I),
    7QALIRE(I),AONHND(I),BONHND(I),OONHND(I),
    8DLROH(I),NQDR(I),PVINP(I),PEELNP(I),QAR(I),PA(I),CAR(I),DEVN(I),
    9NOAR(I),OPER(I),ISPEC(I),GVN(I),WMDR(I),DAYSCL(I),ISEQ(I),RECS(I)
101  FORMAT(A6,A3,A6,A1,A2,A1,2I2,2I4,14X,2I4,3X,I4,I3,2I2,2I1,
    13I4,14X,3I4,I2,I4,2I3,I2,I4,I3,2I4,2X,I4,3X,I4,I3,2I2,I5,5X,
    2F5,2,15X,I6,6X,2A12,5I6,4X,3I6,A12,I2,I2,7X,2I5,5X,A5,2X,I6,2X,I4,2X,
    3F4,0,29X,I2,A2,I1,A4,F5,2,I4,2I3)
    EPA(I)=(FLOAT(CAR(I))/(FLOAT(PA(I))+.0001))*100.0
    IF(EPA(I).GT. 100.0) EPA(I)=100.0
    KCOUNT=KCOUNT+1
    ISTRAT(I)=0
C    DELETE NEXT LINE FROM DLA-WIDE MODEL
C    IF(DCASR(I).NE. ADCASR) GOTO 190
    IF(QAORG(I).EQ. ' ') GOTO 190
    IF(FSCM(I).EQ. ' ') GOTO 190
    IF((TYP(I).NE. 'N') .AND. (TYP(I).NE. 'R')) GOTO 190
    IF(CMDTY(I).GT. 'ZZ') GOTO 190
    IF(CMDTY(I).EQ. 'Y1') GOTO 190
    IF((PVN(I).NE. 'A') .AND. (PVN(I).NE. 'B') .AND. (PVN(I).NE.
    1'C')) GOTO 190
    IF(RECS(I).LT. MONTHS) GOTO 190
    IF(YEAR(I).GT. ENDYR) GOTO 190
    IF((YEAR(I).EQ. ENDYR) .AND. (MONTH(I).GT. ENDMO)) GOTO 190
    IF (YEAR(I).LT. ICUTYR) GOTO 190
    ITOTHR=PLANHR(I)+PVIHR(I)+PEHR(I)+PRHR(I)+QDRHR(I)+TVLHR(I)+
    1TNGHR(I)+FMSHR(I)+ADMNHR(I)+WDHR(I)+INTHR(I)+REINHR(I)+MRBHR(I)+
    2ECPHR(I)+MDRHR(I)+MTGHR(I)+POHR(I)
    IF((AVLHR(I).LE. 2) .AND. (WMDR(I).EQ. 0.0)) GOTO 190
C    JTOTHR=PVIHR(I)+PEHR(I)+PRHR(I)+QDRHR(I)+FMSHR(I)+WDHR(I)+
C    1INTHR(I)+REINHR(I)+MRBHR(I)+ECPHR(I)+MDRHR(I)+POHR(I)+PLANHR(I)
C    IF((ITOTHR.LE. 0) .AND. (WMDR(I).EQ. 0.0)) GOTO 190
C    IF(JTOTHR.EQ. 0) IPRNT(I)=1
C
C CHECK TO SEE IF FACILITY IS STILL UNDER COAP POLICY.
C IF IT IS, REINTERPRET DATA .THIS IS ONLY NEEDED DURING TRANSITION.
C
C    IF((GVN(I)(1:1).NE. 'E') .OR. (GVN(I)(2:2).NE. 'E') .OR.
C    1(GVN(I)(3:3).NE. 'E') .OR. (GVN(I)(4:4).NE. 'E')) THEN
C        DEVN(I)= REAL(WD(I))*40
C        WD(I)=(WD(I)*3)/5
C    ENDIF
C    IF RECORD SURVIVES EDIT CHECK, ASSIGN A STRAT ID NUMBER
C        LCOUNT=LCOUNT + 1
C        CALL STRAT(I,TYP,CMDTY,PVN,OPER,NOAR,ISTRAT,ISEQ,RECS,AVLHR)
C    FURTHER BREAKOUT RESIDENT FACILITIES TO THE SECOND COMMODITY ALPHA.
C    BREAKOUT MAINTENANCE FACILITIES TO THE FIRST COMMODITY ALPHA.
C        CALL PEERGP(I,ISTRAT,IPEER,CMDTY,PVN,NOAR)
C    WRITE NEW RECORD TO DISK FOR FURTHER USE
    WRITE(2,150) DCASR(I),QAORG(I),FSCM(I),TYP(I),CMDTY(I),PVN(I),
    1MONTH(I),YEAR(I),PLANHR(I),LOTINS(I),PVIHR(I),
    2PEHR(I),AQDR(I),BQDR(I),CQDR(I),DQDR(I),EQDR(I),
    3ADMNHR(I),SHIPMT(I),WD(I),INTHR(I),REINHR(I),VISIT(I),
    4ECP(I),MTGHR(I),MRB(I),PCO(I),CAO(I),SI(I),EPA(I),DEVN(I),
    5CONTR(I),DLRIN(I),DLROUT(I),ACNTRT(I),BCNTRT(I),OCNTRT(I),QALIIN
    6(I),QALIRE(I),AONHND(I),BONHND(I),OONHND(I),DLROH(I),NQDR(I),
    7PVINP(I),PEELNP(I),QAR(I),NOAR(I),OPER(I),WMDR(I),

```

```

      8DAYSCL(1),ISEQ(1),RECS(1),ISTRAT(1),IPEER(1),IPRNT(1)
150  FORMAT(A6,A3,A6,A1,A2,A1,2I2,4I4,I3,2I2,2I1,2I4,I2,2I3,I2,I3,2I4,
      .12I2,3F6.2,I6,2A12,8I6,A12,I2,2I5,A5,I2,A2,F5.2,I4,2I3,2I4,I1)
      GO TO 195
190  NCOUNT=NCOUNT+1
195  CONTINUE
      GO TO 100
C
C      EXIT MODULE
C
202  WRITE(6,203) KCOUNT,LCOUNT,NCOUNT
203  FORMAT(5X,'RECORDS READ',1X,I9,10X,'RECORDS WRITTEN ',I9,10X,'RECORDS
      1RDS SKIPPED ',I9)
C
C      ***THIS MODULE IS RUN TO ASSIGN A DEGREE OF DIFFICULTY
C      INDEX TO EACH FACILITY PRIOR TO TOPSIS PROCESSING.
C      PRELIMINARY COMPUTATIONS ARE MADE TO DEVELOP ATTRIBUTES.
C      TOPSIS COMPUTATIONS ARE GENERATED. RED FLAGS ARE IDENTIFIED
C      AND SCORED. OVERALL EFFECTIVENESS IS COMPUTED.
C
C      *****
C      ***THIS SECTION SELECTS DATA RECORDS FROM THE INPUT DISK FILE,
C      COMPARES FSCM WITH CONTRACTOR IMPROVEMENT PROGRAM FILE, AND
C      COMPUTES AND INDEX BASED ON CIP, MDRS AND SEVERE QDRS.
C      CODES ASSIGNED ARE 1-4 AS FOLLOWS: 1 IS A PROBLEM RESIDENT
C      FACILITY, 2 IS A NORMAL RESIDENT, 3 IS A PROBLEM NONRESIDENT AND
C      4 IS A NORMAL NONRESIDENT.
C      ***STEP 1 ***
C      ***INITIALIZE PARAMETERS
C
      REWIND 2
      IFAC=0
      JFAC=0
      ICIPNO=0
C
C      SET THE IDEAL AND NEGATIVE IDEAL CONDITIONS AND WEIGHT FACTORS
      CALL CORNER(AIDEAL,AWORST,BIDEAL,BWORST,WEIGHT)
C
C      READ THE MASTER DATA FILE
C
      DO 640 I=1,999
        READ (10,641) A(I,1),A(I,2),A(I,3),A(I,4),A(I,5),A(I,6),A(I,7),
        1A(I,8),A(I,9),A(I,10),A(I,11),A(I,12),A(I,13),A(I,14),A(I,15),
        2A(I,16),A(I,17),A(I,18),A(I,19),A(I,20),A(I,21),A(I,22),A(I,23),
        3A(I,24),A(I,25),A(I,26),A(I,27),A(I,28),A(I,29),A(I,30),A(I,31),
        4A(I,32),A(I,33),A(I,34),A(I,35),A(I,36),A(I,37),A(I,38)
641  FORMAT(3X,2F7.2,2F6.2,4F7.2,8F6.2,2F5.2,2F7.2,2F6.2,6F5.2,2F7.2,
        12F6.2,2F7.2,2F5.2,2F7.2)
640  CONTINUE
C
C      CHECK TO SEE IF CONTRACTOR ALERT OPTION IS ACTIVE
C
      IF (ICIP.EQ. 1) GOTO 674
C      READ THE CONTRACTOR ALERT FILE
      DO 671 I=1,2000
        ICIPNO=ICIPNO+1
        READ(8,672,END=673) BFSCM(I)
672  FORMAT(4X,A5,81X)
671  CONTINUE
C      IF THIS LOOP IS DONE THE ENTIRE FILE EXCEEDS THE ARRAY LENGTH
      GO TO 675
673  ICIPNO=I-1
C      SET DEFAULTS TO ZERO
674  DO 670 I=1,120
        JCIP=0
        DEGREE(I)=0
670  CONTINUE
C      ***READ FIRST RECORD FOR FACILITY***
600  READ(2,601,END=666) DCASR(1),QAORG(1),FSCM(1),TYP(1),CMDTY(1),
      1PVN(1),MONTH(1),YEAR(1),PLANHR(1),LOTINS(1),PVIHR(1),PEHR(1),
      2AQDR(1),BQDR(1),CQDR(1),DQDR(1),EQDR(1),ADMNHR(1),SHIPMT(1),
      3WD(1),INTHR(1),REINHR(1),VISIT(1),ECP(1),MTGHR(1),MRB(1),PCO(1),
      4CAD(1),
      5SI(1),EPA(1),DEVN(1),CONTR(1),DLRIN(1),DLROUT(1),ACNTRT(1),
      6BCNTRT(1),OCNTRT(1),QALIIN(1),QALIRE(1),AONHND(1),BONHND(1),
      7OONHND(1),DLROH(1),NQDR(1),PVINP(1),PEELNP(1),QAR(1),
      8NQAR(1),OPER(1),WMDR(1),DAYSCL(1),ISEQ(1),RECS(1),ISTRAT(1),
      9IPEER(1),IPRNT(1)
601  FORMAT(A6,A3,A6,A1,A2,A1,2I2,4I4,I3,2I2,2I1,2I4,I2,2I3,I2,I3,2I4,
      12I2,3F6.2,I6,2A12,8I6,A12,I2,2I5,A5,I2,A2,F5.2,I4,2I3,2I4,I1)

```

C	JFAC COUNTS THE FACILITIES THAT REACH THIS STEP.KB COUNTS THE	02400002
C	NUMBER OF RECORDS FOR A GIVEN FACILITY	02410002
	JFAC=JFAC+1	02420002
	KB=1	02430002
	IF(ICIP .EQ. 1) GOTO 604	02440002
C	IF THE ALERT OPTION IS ACTIVE, CHECK TO SEE IF FSCM IS ON ALERT	02450002
	CALL CIP(KB,FSCM,JCIP,ICIPND)	02460099
C	READ ADDITIONAL RECORDS FOR THE FSCM	02470002
604	DO 650 KB=2,120	02480002
	READ(2,601,END=666) DCASR(KB),QAORG(KB),FSCM(KB),TYP(KB),CMDTY(KB),	02490002
	PVN(KB),MONTH(KB),YEAR(KB),PLANHR(KB),LOTINS(KB),PVIHR(KB),	02500002
	2PEHR(KB),AODR(KB),BODR(KB),CQDR(KB),DQDR(KB),EQDR(KB),ADMNHR(KB),	02510042
	3SHIPMT(KB),WD(KB),INTHR(KB),REINHR(KB),VISIT(KB),ECP(KB),	02520042
	4MTGHR(KB),MRB(KB),PCO(KB),CAO(KB),SI(KB),EPA(KB),DEVN(KB),	02530099
	5CONTR(KB),DLRIN(KB),DLROUT(KB),ACNTRT(KB),BCNTRT(KB),OCNTRT(KB),	02540042
	6QALIIN(KB),QALIRE(KB),AONHND(KB),BONHND(KB),OONHND(KB),DLROH(KB),	02550042
	7NQDR(KB),PVINP(KB),PEELNP(KB),QAR(KB),NQAR(KB),	02560099
	8OPER(KB),WMDR(KB),DAYSCL(KB),ISEQ(KB),RECS(KB),ISTRAT(KB),	02570042
	9IPEER(KB),IPRNT(KB)	02580053
	KA=KB-1	02590002
C	STOP READING IF THIS IS THE LAST RECORD FOR FSCM OR A NEW FSCM	02610002
C	APPEARS	02620002
	IF (ISEQ(KB) .EQ. RECS(KB)) GOTO 652	02630002
	IF (FSCM(KB) .GT. FSCM(KA)) GOTO 651	02640002
650	CONTINUE	02650002
	GO TO 678	02660002
651	NCTL=1	02670002
	GO TO 654	02680002
652	NCTL=0	02690002
	KA=KA+1	02700002
654	IF(KA .LT. MONTHS) GOTO 662	02710002
C	IF ENOUGH DATA, ASSIGN DEGREE OF DIFFICULTY INDEX TO FSCM ARRAY.	02720002
C	THEN PREPARE ARRAY FOR TOPSIS BY COMPUTING INDICATORS.	02730002
	DO 660 L1=1,KA	02740002
	ONHAND(L1)=AONHND(L1)+BONHND(L1)+OONHND(L1)	02750026
	CALL DIFF(TYP,CQDR,DQDR,EQDR,WMDR,L1,JCIP,DEGREE,IPEER)	02760060
	IF (LONG .EQ. 0) GOTO 660	02770002
	WRITE(4,661) DCASR(L1),QAORG(L1),FSCM(L1),TYP(L1),CMDTY(L1),	02780002
	1PVN(L1),MONTH(L1),YEAR(L1),PLANHR(L1),LOTINS(L1),PVIHR(L1),	02790002
	2PEHR(L1),AODR(L1),BODR(L1),CQDR(L1),DQDR(L1),EQDR(L1),ADMNHR(L1),	02800002
	3SHIPMT(L1),WD(L1),INTHR(L1),REINHR(L1),VISIT(L1),ECP(L1),	02810002
	4MTGHR(L1),MRB(L1),PCO(L1),CAO(L1),SI(L1),EPA(L1),DEVN(L1),	02820099
	5CONTR(L1),DLRIN(L1),DLROUT(L1),ACNTRT(L1),BCNTRT(L1),OCNTRT(L1),	02830002
	6QALIIN(L1),QALIRE(L1),AONHND(L1),BONHND(L1),OONHND(L1),DLROH(L1),	02840002
	7NQDR(L1),PVINP(L1),PEELNP(L1),NQAR(L1),OPER(L1),WMDR(L1),	02850099
	8DAYSCL(L1),ISEQ(L1),RECS(L1),IPEER(L1),L1,KA,JCIP,DEGREE(L1)	02860099
661	FORMAT(A6,A3,A6,A1,A2,A1,2I2,4I4,I3,2I2,2I1,2I4,I2,2I3,I2,I3,2I4,	02870002
	12I2,3F6,2,I6,2A12,8I6,A12,I2,2I5,I2,A2,F5,2,I4,2I3,I4,4I3)	02880099
660	CONTINUE	02890002
C	COUNT THE FACILITIES THAT HAVE SURVIVED.	02900002
	IFAC=IFAC+1	02910002
C	COMPUTE TOPSIS VALUES FOR FSCM	02920002
	CALL PREPIN(KA,AQDR,BQDR,CQDR,DQDR,EQDR,WD,ECP,MRB,EPA,DEVN,WMDR,	02930099
	1TOPCA,CARATE,WDRATE,ECPRAT,AMRBRA,EPARAT,DEVNRA,WMDRRA,STARTM,	02940099
	2STARTY,ENDMD,ENDYR,FSCM,MONTH,YEAR,TOPWD,TOPECP,TOPMRB,TOPEPA,	02950002
	3TOPDEV,TOPMDR,DCASR,QAORG,DEGREE,AIDEAL,AWORST,BIDEAL,BWORST,	02960099
	4ISTRAT,WEIGHT,REINHR,INTHR,MTGHR,SHIPMT,LOTINS,VISIT,PVN,TYP,	02970002
	5PEHR,CONTR,PLANHR,DAYSCL,PCO,CAO,NQDR,SI,ADMNHR,PVINP,	02980099
	6PEELNP,QALIIN,QALIRE,LONG,TOFSCR,PVIHR,ONHAND,QAR,IPRNT,IPEER,	02990095
	7CMDTY,NQAR)	03000099
C	BEGIN ANOTHER FSCM.	03010002
C	EITHER READ A NEW RECORD OR MOVE LAST RECORD INTO FIRST POSITION	03020002
662	IF(NCTL .EQ. 0) GOTO 600	03030002
	CALL NEWFAC(KB,DCASR,QAORG,FSCM,TYP,CMDTY,PVN,MONTH,YEAR,PLANHR,	03040002
	1LOTINS,PEHR,AQDR,BQDR,CQDR,DQDR,EQDR,ADMNHR,SHIPMT,WD,INTHR,	03050002
	2REINHR,VISIT,ECP,MTGHR,MRB,PCO,CAO,	03060094
	3SI,EPA,DEVN,CONTR,DLRIN,DLROUT,	03070099
	4ACNTRT,BCNTRT,OCNTRT,QALIIN,QALIRE,AONHND,BONHND,OONHND,DLROH,	03080092
	5NQDR,PVINP,PEELNP,NQAR,OPER,WMDR,DAYSCL,ISEQ,RECS,ISTRAT,	03090099
	6JCIP,ICIPND,ICIP,PVIHR,QAR,IPRNT,IPEER)	03100098
	JFAC=JFAC+1	03110002
	GO TO 604	03120002
C	ABNORMAL TERMINATION	03130002
675	WRITE(6,676) ICIPNO	03140002
676	FORMAT(5X,'EXCESS RECORDS ON CONTRACTOR IMPROVEMENT FILE,OVER',I6)	03150002
678	WRITE(6,679)	03160002
679	FORMAT(2X,'ERROR DETECTED. VERIFY INPUT FILE SORTED BY FSCM')	03170002
C	NORMAL TERMINATION	03180002
666	WRITE(6,668) JFAC,IFAC	03190002
668	FORMAT(5X,'FACILITIES IN ',I9,5X,'FACILITIES OUT ',I9)	03200002

WRITE(6,677) ICIPND	03210002
677 FORMAT(5X,'CIP FACILITIES',I6)	03220002
STOP	03230002
END	03240002
C	03250002
SUBROUTINE TIMCHK(MONTH,IYEAR,JERROR)	03260002
C CHECKS TIME VALUES ENTERED FOR INCONSISTENCY	03270002
IF(IYEAR.LT. 84) GOTO 300	03280002
IF(MONTH.LT. 1) GOTO 300	03290002
IF(MONTH.GT. 12) GOTO 300	03300002
JERROR=0	03310002
GO TO 310	03320002
300 JERROR=1	03330002
310 RETURN	03340002
END	03350002
C	03360002
SUBROUTINE STRAT(I,TYP,CMDTY,PVN,OPER,NQAR,JJ,ISEQ,RECS,AVLHR)	03370099
C	03380002
C ASSIGNS A STRATIFICATION ID NUMBER TO EACH RECORD. STRAT ID	03390002
C IS USED TO GROUP SIMILAR FACILITIES. THE NUMBER RANGES INITIALLY	03400002
C FROM 1-742. ID VALUES 1-500 ARE USED FOR NONRESIDENT FACILITIES,	03410046
C RESIDENT ARE 501-740. REPAIR/OVERHAUL FACILITIES ARE 386,741,	03420046
C AND 742 FOR NONRESIDENT, SMALL RESIDENT AND LARGE RESIDENT RESP.	03430046
C NONASSIGNED FACILITIES ARE PUT IN GROUP 385.	03440046
C	03450002
CHARACTER TYP(120)*1,CMDTY(120)*2,PVN(120)*1,OPER(120)*2	03460002
INTEGER NQAR(120),JJ(120),AVLHR(120),ISEQ(120),RECS(120)	03470099
C	03480002
IF((OPER(I).EQ. 'C') .OR. (OPF(I).EQ. 'C') .OR.	03490067
1 (CMDTY(I).EQ. 'A5')) GOTO 59	03500068
C KK IS A DORO FORTRAN UNIQUE VALUE L 1-16 REPRESENTING THE	03510002
C 16 COMMODITY CODES IN DLAM 8200.2	03520002
KK=(ICHAR(CMDTY(I))(1:1))-192	03530002
IF(KK.GT. 33) GOTO 501	03540002
IF(KK.GT. 9) GOTO 502	03550002
GO TO 504	03560002
501 KK=KK-15	03570002
GO TO 504	03580002
502 KK=KK-7	03590002
504 IF (KK.EQ. 16) GOTO 515	03600002
IF (KK.GT. 21) GOTO 516	03610002
IF (KK.GT. 18) GOTO 517	03620002
IF (KK.GT. 10) GOTO 518	03630002
IF (KK.GT. 6) GOTO 519	03640002
GO TO 520	03650002
515 KK=12	03660002
GO TO 520	03670002
516 KK=KK-8	03680002
GO TO 520	03690002
517 KK=KK-6	03700002
GO TO 520	03710002
518 KK=KK-3	03720002
GO TO 520	03730002
519 KK=KK-1	03740002
520 IF (KK.GT. 16) GOTO 599	03750002
C SPLIT RESIDENT FROM NONRESIDENT	03760099
IF(TYP(I).EQ. 'N') GOTO 550	03770002
C RESIDENT ID VALUES DEPEND ON COMMODITY(16), QA PROV(3) AND	03780002
C NUMBER OF QARS(4).QARS IS COMPUTED FOR PRIOR MONTHS	03790099
IF(ISEQ(I).LT. RECS(I)) NQAR(I)=INT(FLOAT(AVLHR(I)-36)/14)+10	03800099
C COMBINE SERVICE WITH GENERAL COMMODITIES	03810099
IF(KK.EQ. 13) KK=6	03820099
KKK=KK*15-15	03830002
C QA PROV IS EITHER A OR B OR C (MIL Q, MIL I OR OTHER)	03840002
IF (PVN(I).EQ. 'A') GOTO 526	03850002
IF (PVN(I).EQ. 'B') GOTO 528	03860002
LL=10	03870002
GO TO 530	03880002
526 LL=0	03890002
GO TO 530	03900002
528 LL=5	03910002
530 IF (NQAR(I).LE. 2) GOTO 532	03920002
IF (NQAR(I).LE. 7) GOTO 534	03930002
IF (NQAR(I).LE. 20) GOTO 536	03940002
MM=5	03950002
GO TO 540	03960002
532 MM=1	03970002
GO TO 540	03980002
534 MM=2	03990002
GO TO 540	04000002

536	MM=3	04010002
540	JJ(I)=KKK+LL+MM+500	04020046
	GO TO 599	04030002
C	ASSIGN STRAT ID TO NONRESIDENT FACILITY	04040002
	VALUES RANGE FROM 1-500 DEPENDING ON 2 ALPHA COMMODITY CODE	04050046
C	IN DLAM 8200.2(16*8) AND QA PROV(3)	04060002
550	MMM=KK*24 24	04070046
	NN=0	04080002
C	COMBINE ALL MIL-Q AT 1 ALPHA FOR C&T, MARINE, CHEMICAL,	04090067
C	NUCLEAR, PETROLEUM, SERVICE, VEHICLES AND MISSILES/SPACE	04100035
	IF((PVN(I) .EQ. 'A') .AND. ((KK .EQ. 3) .OR. (KK .EQ. 4) .OR.	04110035
	1 (KK .EQ. 7) .OR. (KK .EQ. 11) .OR.	04120067
	2 (KK .EQ. 12) .OR. (KK .EQ. 13) .OR. (KK .EQ. 14) .OR.	04130037
	3 (KK .EQ. 16))) GOTO 552	04140037
	NN=(ICHAR(CMDTY(I)(2:2))-240)*3-3	04150002
	IF(PVN(I) .EQ. 'A') GOTO 552	04160018
	IF(PVN(I) .EQ. 'B') GOTO 554	04170002
551	NNN=3	04180002
	GO TO 556	04190022
552	NNN=1	04200002
	GO TO 556	04210002
554	NNN=2	04220002
556	JJ(I)=MMM+NN+NNN	04230002
	IF((JJ(I) .GE. 13) .AND. (JJ(I) .LE. 15)) GOTO 598	04240068
	IF((JJ(I) .GE. 76) .AND. (JJ(I) .LE. 78)) GOTO 598	04250068
C	COMBINE DIFFERENT STRATA	04260002
	CALL STRAT2(I,JJ)	04270002
	GO TO 599	04280002
C	ASSIGN STRAT ID TO MAINTENANCE FACILITIES	04290002
598	JJ(I)=386	04300046
	IF((TYP(I) .EQ. 'R') .AND. (NOAR(I) .GE. 8)) JJ(I)=742	04310057
	IF((TYP(I) .EQ. 'R') .AND. (NOAR(I) .LT. 8)) JJ(I)=741	04320057
599	IF(JJ(I) .EQ. 0) JJ(I)=385	04330046
	RETURN	04340002
	END	04350002
	SUBROUTINE STRAT2(I,JJ)	04360002
C	THIS SUBROUTINE COMBINES NONRESIDENT STRATA	04370002
	INTEGER JJ(120)	04380002
C	**COMBINE H1 AND H2*****	04390018
	IF((JJ(I) .EQ. 149) .OR. (JJ(I) .EQ. 150)) THEN	04400067
	JJ(I)=JJ(I)-3	04410002
	RETURN	04420002
	ENDIF	04430002
C	**COMBINE E2 AND E6 FOR MIL-Q ONLY *****	04440067
	IF(JJ(I) .EQ. 100) THEN	04450067
	JJ(I)=112	04460067
	RETURN	04470035
	ENDIF	04480035
C	**COMBINE G2 AND G7 AND G8 FOR MIL-Q ONLY *****	04490067
	IF((JJ(I) .EQ. 124) .OR. (JJ(I) .EQ. 139)) THEN	04500068
	JJ(I)=142	04510067
	RETURN	04520067
	ENDIF	04530067
C	**COMBINE C5 AND C6 FOR MIL-I ONLY *****	04540035
	IF(JJ(I) .EQ. 62) THEN	04550046
	JJ(I)=65	04560046
	RETURN	04570002
	ENDIF	04580002
C	**COMBINE K4 AND K7 EXCEPT FOR MIL-I *****	04590035
	IF((JJ(I) .EQ. 178) .OR. (JJ(I) .EQ. 180)) THEN	04600046
	JJ(I)=JJ(I)+9	04610035
	RETURN	04620035
	ENDIF	04630035
C	**COMBINE W4 AND W6 EXCEPT FOR MIL-I *****	04640035
	IF((JJ(I) .EQ. 346) .OR. (JJ(I) .EQ. 348)) THEN	04650046
	JJ(I)=JJ(I)+6	04660035
	RETURN	04670035
	ENDIF	04680035
C	**COMBINE P5 AND P6 MIL-I ONLY *****	04690035
	IF(JJ(I) .EQ. 281) THEN	04700046
	JJ(I)=278	04710046
	RETURN	04720002
	ENDIF	04730002
C	**COMBINE D3 AND D7 OTHER INSP ONLY *****	04740035
	IF(JJ(I) .EQ. 81) THEN	04750046
	JJ(I)=93	04760046
	RETURN	04770035
	ENDIF	04780035
C	**COMBINE N4 AND N5 OTHER INSP ONLY *****	04790035
	IF(JJ(I) .EQ. 252) THEN	04800046

JJ(I)=255	04810046
RETURN	04820035
ENDIF	04830035
RETURN	04840002
END	04850002
C	04860047
C *** THIS SUBROUTINE TRANSLATES STRAT NUMBERS INTO PEER GROUP NUMBERS	04670047
C RESIDENT FACILITIES ARE BROKEN OUT TO THE SECOND CMDY ALPHA	04880047
SUBROUTINE PEERGP(I,ISTRAT,IPEER,CMDTY,PVN,NQAR)	04890058
INTEGER ISTRAT(120),IPEER(120),NQAR(120)	04900047
CHARACTER CMDTY(120)*2,PVN(120)*1	04910047
C NONRESIDENT FACILITIES ARE NOT CHANGED.	04920067
IPEER(I)=ISTRAT(I)	04930047
IF(ISTRAT(I) .LT. 386) GOTO 10	04940067
C MAINTENANCE FACILITIES ARE BROKEN OUT FURTHER	04950068
IF(IPEER(I) .EQ. 386) THEN	04960068
IF(CMDTY(I)(1:1) .EQ. 'A') THEN	04970068
IPEER(I)=401	04980068
GO TO 10	04990068
ENDIF	05000068
IF(CMDTY(I)(1:1) .EQ. 'D') THEN	05010068
IPEER(I)=404	05020068
GO TO 10	05030068
ENDIF	05040068
IF(CMDTY(I)(1:1) .EQ. 'E') THEN	05050068
IPEER(I)=405	05060068
GO TO 10	05070068
ENDIF	05080068
IF(CMDTY(I)(1:1) .EQ. 'G') THEN	05090068
IPEER(I)=406	05100068
GO TO 10	05110068
ENDIF	05120068
IF(CMDTY(I)(1:1) .EQ. 'K') THEN	05130068
IPEER(I)=408	05140068
GO TO 10	05150068
ENDIF	05160068
IF(CMDTY(I)(1:1) .EQ. 'L') THEN	05170068
IPEER(I)=409	05180068
GO TO 10	05190068
ENDIF	05200068
IF(CMDTY(I)(1:1) .EQ. 'M') THEN	05210068
IF(PVN(I) .EQ. 'A') THEN	05220068
IPEER(I)=410	05230068
GO TO 10	05240068
ENDIF	05250068
IF(PVN(I) .EQ. 'B') THEN	05260068
IF(CMDTY(I) .EQ. 'M3') THEN	05270068
IPEER(I)=412	05280068
GO TO 10	05290068
ELSE	05300068
IPEER(I)=411	05310068
GO TO 10	05320068
ENDIF	05330068
ELSE	05340068
IF(CMDTY(I) .EQ. 'M1') THEN	05350068
IPEER(I)=413	05360068
GO TO 10	05370068
ENDIF	05380068
IF(CMDTY(I) .EQ. 'M2') THEN	05390068
IPEER(I)=414	05400068
GO TO 10	05410068
ENDIF	05420068
IF(CMDTY(I) .EQ. 'M3') THEN	05430068
IPEER(I)=415	05440068
GO TO 10	05450068
ENDIF	05460068
ENDIF	05470068
ENDIF	05480068
IF(CMDTY(I)(1:1) .EQ. 'V') THEN	05490068
IPEER(I)=419	05500068
GO TO 10	05510068
ENDIF	05520068
IF(CMDTY(I)(1:1) .EQ. 'X') THEN	05530068
IPEER(I)=420	05540068
GO TO 10	05550068
ENDIF	05560068
IPEER(I)=400	05570068
GO TO 10	05580068
ENDIF	05590068
IF(IPEER(I) .EQ. 742) THEN	05600067

IPEER(I)=999	05610067
GO TO 10	05620048
ENDIF	05630048
C SMALL RESIDENT MAINTENANCE FACILITIES BY FIRST COMMODITY ALPHA	05640067
IF(IPEER(I) .EQ. 741) THEN	05650067
IF(CMDTY(I) .EQ. 'A5') THEN	05660067
IPEER(I)=980	05670067
GO TO 10	05680067
ENDIF	05690067
IF(CMDTY(I)(1:1) .EQ. 'A') THEN	05700067
IPEER(I)=981	05710067
GO TO 10	05720067
ENDIF	05730067
IF(CMDTY(I)(1:1) .EQ. 'L') THEN	05740067
IPEER(I)=983	05750067
GO TO 10	05760067
ENDIF	05770067
IF(CMDTY(I)(1:1) .EQ. 'M') THEN	05780067
IPEER(I)=984	05790067
GO TO 10	05800067
ENDIF	05810067
IF(CMDTY(I)(1:1) .EQ. 'S') THEN	05820067
IPEER(I)=985	05830067
GO TO 10	05840067
ELSE	05850067
IPEER(I)=982	05860067
GO TO 10	05870067
ENDIF	05880067
ENDIF	05890067
C BREAKOUT AIRCRAFT COMMODITY -- RESERVE GROUPS 501-550	05900067
IF(CMDTY(I)(1:1) .EQ. 'A') THEN	05910048
C THIS SECTION OF CODE IS FOR AIRCRAFT OTHER, PEER GROUPS 541-545	05920048
IF(PVN(I) .EQ. 'C') THEN	05930048
IPEER(I)=IPEER(I)+30	05940048
GO TO 10	05950048
ENDIF	05960048
C THIS SECTION OF CODE IS FOR AIRCRAFT MIL-Q, PEER GROUPS 501-520	05970048
IF(PVN(I) .EQ. 'A') THEN	05980048
IF(NOAR(I) .LT. 8) THEN	05990048
IPEER(I)=IPEER(I)+((ISTRAT(I)-501)*6)+ICHAR(CMDTY(I)(2:2))-241	06000056
GO TO 10	06010048
ELSE	06020048
IF(NOAR(I) .GT. 20) THEN	06030048
IPEER(I)=520	06040048
GO TO 10	06050048
ELSE	06060048
IF((CMDTY(I)(2:2) .EQ. '3') .OR. (CMDTY(I)(2:2) .EQ. '6'))	06070048
1 .OR. (CMDTY(I)(2:2) .EQ. '7')) THEN	06080048
IPEER(I)=517	06090056
GO TO 10	06100048
ELSE	06110048
IF(CMDTY(I)(2:2) .EQ. '1') THEN	06120048
IPEER(I)=515	06130048
GO TO 10	06140048
ELSE	06150048
IF(CMDTY(I)(2:2) .EQ. '2') THEN	06160048
IPEER(I)=516	06170048
GO TO 10	06180048
ELSE	06190048
IPEER(I)=518	06200056
GO TO 10	06210048
ENDIF	06220048
ENDIF	06230048
ENDIF	06240048
ENDIF	06250048
ENDIF	06260048
ELSE	06270048
C THIS SECTION OF CODE IS FOR AIRCRAFT MIL-I, PEER GROUPS 521-540	06280051
IF(NOAR(I) .GT. 2) THEN	06290048
IPEER(I)=IPEER(I)+30	06300048
GO TO 10	06310048
ELSE	06320048
IF((CMDTY(I)(2:2) .EQ. '3') .OR. (CMDTY(I)(2:2) .EQ. '6'))	06330048
1 THEN	06340048
IPEER(I)=523	06350048
GO TO 10	06360048
ELSE	06370048
IPEER(I)=IPEER(I)+((ISTRAT(I)-491)+ICHAR(CMDTY(I)(2:2))-241	06380051
GO TO 10	06390048
ENDIF	06400048

ENDIF	06410048
ENDIF	06420048
ENDIF	06430048
C BREAKOUT MUNITIONS COMMODITY -- RESERVE GROUPS 551-575	06440054
IF(CMDTY(I)(1:1) .EQ. 'B') THEN	06450054
C THIS SECTION OF CODE IS FOR MUNITIONS MIL-I, PEER GROUPS 566-570	06460054
C THIS SECTION OF CODE IS FOR MUNITIONS OTHER, PEER GROUPS 571-575	06470054
IF((PVN(I) .EQ. 'B') .OR. (PVN(I) .EQ. 'C')) THEN	06480054
IPEER(I)=IPEER(I)+45	06490054
GO TO 10	06500054
ENDIF	06510054
C THIS SECTION OF CODE IS FOR MUNITIONS MIL-Q, PEER GROUPS 551-565	06520054
IF(NQAR(I) .LT. 8) THEN	06530054
IPEER(I)=IPEER(I)+(ISTRAT(I)-516)*5+ICHAR(CMDTY(I)(2:2))-241+3506540054	
IF(CMDTY(I)(2:2) .EQ. '5') IPEER(I)=IPEER(I)-2	06550054
IF(NQAR(I) .GT. 2) IPEER(I)=IPEER(I)-1	06560054
GO TO 10	06570054
ELSE	06580054
IF(NQAR(I) .GT. 20) THEN	06590054
IPEER(I)=565	06600054
GO TO 10	06610054
ELSE	06620054
IPEER(I)=563	06630054
GO TO 10	06640054
ENDIF	06650054
ENDIF	06660054
ENDIF	06670054
C BREAKOUT MUNITIONS COMMODITY -- RESERVE GROUPS 551-575	06680056
IF(CMDTY(I)(1:1) .EQ. 'B') THEN	06690056
C THIS SECTION OF CODE IS FOR MUNITIONS MIL-I, PEER GROUPS 566-570	06700056
C THIS SECTION OF CODE IS FOR MUNITIONS OTHER, PEER GROUPS 571-575	06710056
IF((PVN(I) .EQ. 'B') .OR. (PVN(I) .EQ. 'C')) THEN	06720056
IPEER(I)=IPEER(I)+45	06730056
GO TO 10	06740056
ENDIF	06750056
C THIS SECTION OF CODE IS FOR MUNITIONS MIL-Q, PEER GROUPS 551-565	06760056
IF(NQAR(I) .LT. 8) THEN	06770056
IPEER(I)=IPEER(I)+(ISTRAT(I)-516)*5+ICHAR(CMDTY(I)(2:2))-241+3506780056	
IF(CMDTY(I)(2:2) .EQ. '5') IPEER(I)=IPEER(I)-2	06790056
IF(NQAR(I) .GT. 2) IPEER(I)=IPEER(I)-1	06800056
GO TO 10	06810056
ELSE	06820056
IF(NQAR(I) .GT. 20) THEN	06830056
IPEER(I)=565	06840056
GO TO 10	06850056
ELSE	06860056
IPEER(I)=563	06870056
GO TO 10	06880056
ENDIF	06890056
ENDIF	06900056
ENDIF	06910056
C BREAKOUT C&T COMMODITY -- RESERVE GROUPS 576-600	06920056
IF(CMDTY(I)(1:1) .EQ. 'C') THEN	06930056
C THIS SECTION OF CODE IS FOR C&T OTHER, PEER GROUPS 591-595	06940056
IF(PVN(I) .EQ. 'C') THEN	06950056
IPEER(I)=IPEER(I)+50	06960056
GO TO 10	06970056
ENDIF	06980056
C THIS SECTION OF CODE IS FOR C&T MIL-Q, PEER GROUPS 576-580	06990056
IF(PVN(I) .EQ. 'A') THEN	07000056
IPEER(I)=IPEER(I)+45	07010056
GO TO 10	07020056
ELSE	07030056
C THIS SECTION OF CODE IS FOR C&T MIL-I, PEER GROUPS 581-590	07040056
IF(NQAR(I) .GT. 2) THEN	07050056
IPEER(I)=IPEER(I)+50	07060056
GO TO 10	07070056
ELSE	07080056
IF((CMDTY(I)(2:2) .EQ. '1') .OR. (CMDTY(I)(2:2) .EQ. '2'))	07090056
1 .OR. (CMDTY(I)(2:2) .EQ. '5') .OR. (CMDTY(I)(2:2) .EQ. '6'))	07100056
2 THEN	07110056
IPEER(I)=581	07120056
GO TO 10	07130056
ELSE	07140056
IF(CMDTY(I)(2:2) .EQ. '3') THEN	07150056
IPEER(I)=582	07160056
GO TO 10	07170056
ELSE	07180056
IPEER(I)=583	07190056
GO TO 10	07200056

ENDIF	07210056
ENDIF	07220056
ENDIF	07230056
ENDIF	07240056
ENDIF	07250056
C BREAKOUT MARINE COMMODITY -- RESERVE GROUPS 601-625	07260057
IF(CMDTY(I)(1:1) .EQ. 'D') THEN	07270057
C THIS SECTION OF CODE IS FOR MARINE OTHER, PEER GROUPS 621-625	07280057
IF(PVN(I) .EQ. 'C') THEN	07290057
IPEER(I)=IPEER(I)+65	07300059
GO TO 10	07310057
ENDIF	07320057
C THIS SECTION OF CODE IS FOR MARINE MIL-Q, PEER GROUPS 601-605	07330057
IF(PVN(I) .EQ. 'A') THEN	07340057
IPEER(I)=IPEER(I)+55	07350057
GO TO 10	07360057
ELSE	07370057
C THIS SECTION OF CODE IS FOR MARINE MIL-I, PEER GROUPS 606-620	07380057
IF(NOAR(I) .GT. 2) THEN	07390057
IPEER(I)=IPEER(I)+60	07400057
GO TO 10	07410057
ELSE	07420057
IF(CMDTY(I)(2:2) .EQ. '5') THEN	07430057
IPEER(I)=607	07440057
GO TO 10	07450057
ELSE	07460057
IPEER(I)=606	07470057
GO TO 10	07480057
ENDIF	07490057
ENDIF	07500057
ENDIF	07510057
ENDIF	07520057
C BREAKOUT ELECTRICAL COMMODITY -- RESERVE GROUPS 626-650	07530057
IF(CMDTY(I)(1:1) .EQ. 'E') THEN	07540057
C THIS SECTION OF CODE IS FOR ELECTRICAL OTHER, PEER GROUPS 646-650	07550057
IF(PVN(I) .EQ. 'C') THEN	07560057
IPEER(I)=IPEER(I)+75	07570057
GO TO 10	07580057
ENDIF	07590057
C THIS SECTION OF CODE IS FOR ELECTRICAL MIL-Q, PEER GROUPS 626-635	07600057
IF(PVN(I) .EQ. 'A') THEN	07610057
IF(NOAR(I) .LT. 8) THEN	07620057
IF(NOAR(I) .LT. 3) THEN	07630057
IF(CMDTY(I)(2:2) .EQ. '1') THEN	07640057
IPEER(I)=626	07650057
GO TO 10	07660057
ELSE	07670057
IF(CMDTY(I)(2:2) .EQ. '5') THEN	07680057
IPEER(I)=628	07690057
GO TO 10	07700057
ELSE	07710057
IPEER(I)=627	07720057
GO TO 10	07730057
ENDIF	07740057
ENDIF	07750057
ELSE	07760057
IF((CMDTY(I)(2:2) .EQ. '1') .OR. (CMDTY(I)(2:2) .EQ. '6'))	07770057
THEN	07780057
IPEER(I)=629	07790057
GO TO 10	07800057
ELSE	07810057
IF((CMDTY(I)(2:2) .EQ. '3') .OR. (CMDTY(I)(2:2) .EQ. '4'))	07820057
THEN	07830057
IPEER(I)=631	07840057
GO TO 10	07850057
ELSE	07860057
IF(CMDTY(I)(2:2) .EQ. '2') THEN	07870057
IPEER(I)=630	07880057
GO TO 10	07890057
ELSE	07900057
IPEER(I)=632	07910057
GO TO 10	07920057
ENDIF	07930057
ENDIF	07940057
ENDIF	07950057
ENDIF	07960057
ELSE	07970057
IPEER(I)=IPEER(I)+70	07980057
GO TO 10	07990057
ENDIF	08000057

ELSE	08010057
C THIS SECTION OF CODE IS FOR ELECTRICAL MIL-I, PEER GROUPS 636-645	08020057
IF(NQAR(I) .GT. 2) THEN	08030057
IPEER(I)=IPEER(I)+75	08040057
GO TO 10	08050057
ELSE	08060057
IF((CMDTY(I)(2:2) .EQ. '1') .OR. (CMDTY(I)(2:2) .EQ. '2'))	08070057
1 .OR. (CMDTY(I)(2:2) .EQ. '4')) THEN	08080057
IPEER(I)=636	08090057
GO TO 10	08100057
ELSE	08110057
IF(CMDTY(I)(2:2) .EQ. '3') THEN	08120057
IPEER(I)=637	08130057
GO TO 10	08140057
ELSE	08150057
IF(CMDTY(I)(2:2) .EQ. '5') THEN	08160057
IPEER(I)=638	08170057
GO TO 10	08180057
ELSE	08190057
IPEER(I)=639	08200057
GO TO 10	08210057
ENDIF	08220057
ENDIF	08230057
ENDIF	08240057
ENDIF	08250057
ENDIF	08260057
ENDIF	08270057
C BREAKOUT GENERAL/SERVICE COMMODITY -- RESERVE GROUPS 651-675	08280057
IF((CMDTY(I)(1:1) .EQ. 'G') .OR. (CMDTY(I)(1:1) .EQ. 'S')) THEN	08290057
C THIS SECTION OF CODE IS FOR GENERAL/SERV OTHER, PEER GROUPS 671-675	08300057
IF(PVN(I) .EQ. 'C') THEN	08310057
IPEER(I)=IPEER(I)+85	08320057
GO TO 10	08330057
ENDIF	08340057
C THIS SECTION OF CODE IS FOR GENERAL/SERV MIL-Q, PEER GROUPS 651-655	08350057
IF(PVN(I) .EQ. 'A') THEN	08360057
IF((CMDTY(I)(2:2) .EQ. '6') .AND. (NQAR(I) .LT. 3)) THEN	08370057
IPEER(I)=654	08380057
GO TO 10	08390057
ELSE	08400057
IPEER(I)=IPEER(I)+75	08410057
GO TO 10	08420057
ENDIF	08430057
ELSE	08440057
C THIS SECTION OF CODE IS FOR GENERAL/SERV MIL-I, PEER GROUPS 656-670	08450057
IF(NQAR(I) .GT. 2) THEN	08460057
IPEER(I)=IPEER(I)+80	08470057
GO TO 10	08480057
ELSE	08490057
IF(CMDTY(I)(2:2) .EQ. '6') THEN	08500057
IPEER(I)=657	08510057
GO TO 10	08520057
ELSE	08530057
IF(CMDTY(I)(2:2) .EQ. '8') THEN	08540057
IPEER(I)=658	08550057
GO TO 10	08560057
ELSE	08570057
IPEER(I)=656	08580057
GO TO 10	08590057
ENDIF	08600057
ENDIF	08610057
ENDIF	08620057
ENDIF	08630057
ENDIF	08640057
C BREAKOUT CHEMICAL COMMODITY -- RESERVE GROUPS 676-700	08650057
IF(CMDTY(I)(1:1) .EQ. 'H') THEN	08660057
C THIS SECTION OF CODE IS FOR ALL CHEMICAL PROVS, PEER GROUPS 676-690	08670057
IPEER(I)=IPEER(I)+85	08680057
GO TO 10	08690057
ENDIF	08700057
C BREAKOUT ELECTRONIC SYSTEMS COMMODITY -- RESERVE GROUPS 701-750	08710057
IF(CMDTY(I)(1:1) .EQ. 'K') THEN	08720057
C THIS SECTION IS FOR ELECTRONIC SYSTEMS OTHER, PEER GROUPS 746-750	08730057
IF(PVN(I) .EQ. 'C') THEN	08740057
IPEER(I)=IPEER(I)+130	08750057
GO TO 10	08760057
ENDIF	08770057
C THIS SECTION IS FOR ELECTRONIC SYSTEMS MIL-Q, PEER GROUPS 701-725	08780057
IF(PVN(I) .EQ. 'A') THEN	08790057
IF(NQAR(I) .LE. 20) THEN	08800099

```

      IPEER(I)=IPEER(I)+((ISTRAT(I)-606)*7)+ICHAR(CMDTY(I)(2:2))-24108810057
1      +95 08820057
      IF((NQAR(I) .GT. 7) .AND. ((CMDTY(I)(2:2) .EQ. '4') .OR.
1      (CMDTY(I)(2:2) .EQ. '6')) IPEER(I)=IPEER(I)-1 08830057
      GO TO 10 08840057
      ELSE 08850057
      IPEER(I)=725 08860057
      GO TO 10 08870057
      ENDIF 08880057
      ELSE 08890057
C THIS SECTION IS FOR ELECTRONIC SYSTEMS MIL-I, PEER GROUPS 726-745 08900057
      IPEER(I)=IPEER(I)+115 08910057
      GO TO 10 08920057
      ENDIF 08930057
      ENDIF 08940057
C BREAKOUT ELECTRONICS COMMODITY -- RESERVE GROUPS 751-800 08950057
      IF(CMDTY(I)(1:1) .EQ. 'L') THEN 08960057
C THIS SECTION OF CODE IS FOR ELECTRONICS OTHER, PEER GROUPS 786-800 08970057
      IF(PVN(I) .EQ. 'C') THEN 08980057
      IPEER(I)=IPEER(I)+155 08990057
      IF((NQAR(I) .LT. 3) .AND. (CMDTY(I)(2:2) .EQ. '4')) IPEER(I)=791 09000057
      GO TO 10 09010057
      ENDIF 09020057
C THIS SECTION OF CODE IS FOR ELECTRONICS MIL-Q, PEER GROUPS 751-770 09030057
      IF(PVN(I) .EQ. 'A') THEN 09040057
      IF(NQAR(I) .LT. 8) THEN 09050057
      IPEER(I)=IPEER(I)+((ISTRAT(I)-621)*5)+ICHAR(CMDTY(I)(2:2))-24109070057
1      +130 09080057
      IF((NQAR(I) .GT. 2) .AND. (CMDTY(I)(2:2) .EQ. '1')) IPEER(I)= 09090057
1      761 09100069
      IF((NQAR(I) .GT. 2) .AND. (CMDTY(I)(2:2) .EQ. '4')) IPEER(I)= 09110057
1      758 09120065
      GO TO 10 09130057
      ELSE 09140057
      IPEER(I)=IPEER(I)+140 09150057
      GO TO 10 09160057
      ENDIF 09170057
      ELSE 09180057
C THIS SECTION OF CODE IS FOR ELECTRONICS MIL-I, PEER GROUPS 771-785 09190057
      IF(NQAR(I) .GT. 2) THEN 09200057
      IPEER(I)=IPEER(I)+150 09210057
      GO TO 10 09220057
      ELSE 09230057
      IF(CMDTY(I)(2:2) .EQ. '3') THEN 09240058
      IPEER(I)=775 09250057
      GO TO 10 09260057
      ELSE 09270057
      IPEER(I)=IPEER(I)+ICHAR(CMDTY(I)(2:2))-241+145 09280057
      GO TO 10 09290057
      ENDIF 09300057
      ENDIF 09310057
      ENDIF 09320057
      ENDIF 09330057
C BREAKOUT MECHANICAL COMMODITY -- RESERVE GROUPS 801-825 09340058
      IF(CMDTY(I)(1:1) .EQ. 'M') THEN 09350058
C THIS SECTION OF CODE IS FOR MECHANICAL OTHER, PEER GROUPS 821-825 09360058
      IF(PVN(I) .EQ. 'C') THEN 09370058
      IPEER(I)=IPEER(I)+175 09380058
      GO TO 10 09390058
      ENDIF 09400058
C THIS SECTION OF CODE IS FOR MECHANICAL MIL-Q, PEER GROUPS 801-810 09410058
      IF(PVN(I) .EQ. 'A') THEN 09420058
      IF(NQAR(I) .LT. 8) THEN 09430058
      IPEER(I)=IPEER(I)+((ISTRAT(I)-636)*3)+ICHAR(CMDTY(I)(2:2))-24109440058
1      +165 09450058
      IF((NQAR(I) .GT. 2) .AND. (CMDTY(I)(2:2) .EQ. '2')) IPEER(I)= 09460058
1      807 09470084
      GO TO 10 09480058
      ELSE 09490058
      IPEER(I)=IPEER(I)+170 09500058
      GO TO 10 09510058
      ENDIF 09520058
      ELSE 09530058
C THIS SECTION OF CODE IS FOR MECHANICAL MIL-I, PEER GROUPS 811-820 09540058
      IF(NQAR(I) .GT. 2) THEN 09550058
      IPEER(I)=IPEER(I)+175 09560058
      GO TO 10 09570058
      ELSE 09580058
      IPEER(I)=IPEER(I)+ICHAR(CMDTY(I)(2:2))-241+170 09590058
      GO TO 10 09600058

```

ENDIF	09610058
ENDIF	09620058
ENDIF	09630058
C BREAKOUT NUCLEAR COMMODITY -- RESERVE GROUPS 826-850	09640058
IF(CMDTY(I)(1:1) .EQ. 'N') THEN	09650058
C THIS SECTION OF CODE IS FOR NUCLEAR OTHER, PEER GROUPS 846-850	09660058
IF(PVN(I) .EQ. 'C') THEN	09670058
IPEER(I)=IPEER(I)+185	09680058
GO TO 10	09690058
ENDIF	09700058
C THIS SECTION OF CODE IS FOR NUCLEAR MIL-Q, PEER GROUPS 826-835	09710058
IF(PVN(I) .EQ. 'A') THEN	09720058
IF(NQAR(I) .LT. 8) THEN	09730058
IF(NQAR(I) .GT. 2) THEN	09740058
IF(CMDTY(I)(2:2) .EQ. '2') THEN	09750058
IPEER(I)=832	09760058
GO TO 10	09770058
ELSE	09780058
IPEER(I)=831	09790058
GO TO 10	09800058
ENDIF	09810058
ELSE	09820058
IF((CMDTY(I)(2:2) .EQ. '1').OR. (CMDTY(I)(2:2) .EQ. '4'))	09830058
1 .OR. (CMDTY(I)(2:2) .EQ. '5')) THEN	09840058
IPEER(I)=826	09850058
GO TO 10	09860058
ELSE	09870058
IF(CMDTY(I)(2:2) .EQ. '2') THEN	09880058
IPEER(I)=827	09890058
GO TO 10	09900058
ELSE	09910058
IPEER(I)=828	09920058
GO TO 10	09930058
ENDIF	09940058
ENDIF	09950058
ENDIF	09960058
ELSE	09970058
IPEER(I)=IPEER(I)+180	09980058
GO TO 10	09990058
ENDIF	10000058
ELSE	10010058
C THIS SECTION OF CODE IS FOR NUCLEAR MIL-I, PEER GROUPS 836-845	10020058
IF(NQAR(I) .GT. 2) THEN	10030058
IPEER(I)=IPEER(I)+185	10040058
GO TO 10	10050058
ELSE	10060058
IF((CMDTY(I)(2:2) .EQ. '1') .OR. (CMDTY(I)(2:2) .EQ. '4'))	10070058
1 .OR. (CMDTY(I)(2:2) .EQ. '5')) THEN	10080058
IPEER(I)=836	10090058
GO TO 10	10100058
ELSE	10110058
IF(CMDTY(I)(2:2) .EQ. '2') THEN	10120058
IPEER(I)=837	10130058
GO TO 10	10140058
ELSE	10150058
IPEER(I)=838	10160058
GO TO 10	10170058
ENDIF	10180058
ENDIF	10190058
ENDIF	10200058
ENDIF	10210058
ENDIF	10220058
C BREAKOUT PETROLEUM COMMODITY -- RESERVE GROUPS 851-875	10230058
IF(CMDTY(I)(1:1) .EQ. 'P') THEN	10240058
C THIS SECTION OF CODE IS FOR ALL PETROLEUM PROVS	10250058
IPEER(I)=IPEER(I)+185	10260058
GO TO 10	10270058
ENDIF	10280058
C BREAKOUT VEHICLE COMMODITY -- RESERVE GROUPS 876-900	10290058
IF(CMDTY(I)(1:1) .EQ. 'V') THEN	10300058
C THIS SECTION OF CODE IS FOR ALL VEHICLE PROVS	10310058
IPEER(I)=IPEER(I)+180	10320058
GO TO 10	10330058
ENDIF	10340058
C BREAKOUT WEAPONS COMMODITY -- RESERVE GROUPS 901-925	10350059
IF(CMDTY(I)(1:1) .EQ. 'W') THEN	10360059
C THIS SECTION OF CODE IS FOR ALL WEAPONS PROVS	10370059
IPEER(I)=IPEER(I)+190	10380059
GO TO 10	10390059
ENDIF	10400059

```

C BREAKOUT MISSILES/SPACE COMMODITY -- RESERVE GROUPS 926-950 10410059
  IF(CMDTY(I)(1:1) .EQ. 'X') THEN 10420059
C THIS SECTION IS FOR MISSILES/SPACE MIL-I&OTHER, PEER GROUPS 941-950 10430059
  IF((PVN(I) .EQ. 'B') .OR. (PVN(I) .EQ. 'C')) THEN 10440059
    IPEER(I)=IPEER(I)+210 10450059
    GO TO 10 10460059
  ENDIF 10470059
C THIS SECTION IS FOR MISSILES/SPACE MIL-O, PEER GROUPS 926-940 10480059
  IF(NOAR(I) .LE. 20) THEN 10490059
    IPEER(I)=IPEER(I)+((ISTRAT(I)-726)*4)+ICHAR(CMDTY(I)(2:2))-241 10500059
  1 +200 10510059
    IF(CMDTY(I)(2:2) .EQ. '4') IPEER(I)=IPEER(I)-1 10520059
    IF(CMDTY(I)(2:2) .EQ. '5') IPEER(I)=IPEER(I)-2 10530059
    GO TO 10 10540059
  ELSE 10550059
    IPEER(I)=940 10560059
    GO TO 10 10570059
  ENDIF 10580059
  ENDIF 10590059
10 RETURN 10600047
END 10610047
C 10620047
C *** THIS SUBROUTINE ASSIGNS A DIFFICULTY INDEX TO A FACILTY OF 1 TO 4. 10630002
SUBROUTINE DIFF(TYP,CQDR,DQDR,EQDR,WMDR,L1,JCIP,DEGREE,IPEER) 10640060
CHARACTER TYP(120)*1 10650002
INTEGER CQDR(120),DQDR(120),EQDR(120),DEGREE(120),IPEER(120) 10660099
REAL WMDR(120),AAVG(120),A(999,38) 10680052
COMMON /RVAR/A 10690002
C RESIDENT ASSIGNMENT IS 1 OR 2. NORMAL IS 2. PROBLEM IS 1. 10700098
C NONRESIDENT ASSIGNMENT IS 3 OR 4 DEPENDING ON ALERT MATCH, MDR 10710002
C ACTIVITY OR CORRECTIVE ACTIONS C D E,NORMAL IS 4. PROBLEM IS 3. 10720098
ASUM=0.0 10730098
AAVG(L1)=0.0 10740002
IBAD=0 10750002
NEWMDR=0 10760098
IF (JCIP .EQ. 1) IBAD=IBAD+1 10770099
IF(L1, GT. 2) THEN 10780099
  IF(CQDR(L1)+CQDR(L1-1)+CQDR(L1-2) .GT. 0) IBAD=IBAD+1 10790099
  IF(EQDR(L1)+EQDR(L1-1)+EQDR(L1-2) .GT. 0) IBAD=IBAD+1 10800099
ELSE 10810099
  IF(CQDR(L1) .GT. 0) IBAD=IBAD+1 10820099
  IF(EQDR(L1) .GT. 0) IBAD=IBAD+1 10830099
ENDIF 10840099
DO 15 KK=1,L1 10850002
  IF (DQDR(KK).GT. 0) IBAD=IBAD+1 10860002
  ASUM=ASUM+WMDR(KK) 10870002
  AAVG(KK)=ASUM/REAL(KK) 10880005
  IF(KK .LT. L1-2) GOTO 15 10890098
  IF(WMDR(KK) .NE. 0.0) NEWMDR=NEWMDR+1 10900098
15 CONTINUE 10910002
IF(TYP(L1) .EQ. 'R') THEN 10920098
  IF(AAVG(L1) .GT. (A(IPEER(L1),35)+A(IPEER(L1),36))) IBAD=IBAD+1 10930098
  DEGREE(L1)=2 10940098
  IF(IBAD .GE. 2) DEGREE(L1)=1 10950098
ELSE 10960098
  IF(NEWMDR GT 0) IBAD=IBAD+1 10970098
  DEGREE(L1)=4 10980098
  IF(IBAD .GE. 2) DEGREE(L1)=3 10990098
ENDIF 11000098
RETURN 11010098
END 11020002
C *** THIS SUBROUTINE COMPARES THE FSCM OF RECORD WITH THE ALERT FILE. 11030002
C IF THERE IS A MATCH A VALUE OF 1 IS ASSIGNED TO JCIP,ELSE IT'S 0. 11040002
SUBROUTINE CIP(J,AFSCM,JCIP,ICIPNO) 11050099
CHARACTER AFSCM(120)*6,BFSCM(2000)*5,FLAG(120)*18 11060099
COMMON /CHTR/BFSCM,FLAG 11080002
DO 10 I=1,ICIPNO 11090002
  IF(AFSCM(J)(2:6) .EQ. BFSCM(I)) GOTO 25 11110099
10 CONTINUE 11120002
GO TO 35 11130002
25 JCIP=1 11140099
GO TO 40 11150098
35 JCIP=0 11160099
40 RETURN 11170002
END 11180002
C *** THIS SUBROUTINE IS CALLED WHEN A NEW FSCM IS READ. 11190002
C THE CONTENTS OF THE NEW FSCM ARE MOVED TO THE FIRST POSITION OF 11200002
C FACILITY ARRAY 11210002
C ***** 11220002
SUBROUTINE NEWFAC(J,DCASR,QAORG,FSCM,TYP,CMDTY,PVN,MONTH,YEAR, 11230002

```

1PLANHR, LOTINS, PEHR, AQDR, BQDR, CQDR, DQDR, EQDR, ADMNHR, SHIPMT, WD, INTHR	11240002
2, REINHR, VISIT, ECP, MTGHR, MRB, PCO, CAO,	11250094
3SI, EPA, DEVN, CONTR, DLRIN, DLROUT,	11260099
4ACNTR1, BCNTR1, UCNTR1, QALIIN, QALIRE, AONHND, BONHND, OONHND, DLROH,	11270092
5NQDR, PVINP, PEELNP, NQAR, OPER, WMDR, DAYSCL, ISEQ, RECS, ISTRAT,	11280099
6JCIP, ICIPNO, ICIP, PVIHR, QAR, IPRNT, IPEER)	11290098
	11300002
C CHARACTER DCASR(120)*6, QAORG(120)*3, FSCM(120)*6, TYP(120)*1,	11310002
1CMDTY(120)*2, PVN(120)*1, OPER(120)*2, BFSCM(2000)*5, QAR(120)*5,	11320099
2FLAG(120)*18	11321099
INTEGER MONTH(120), YEAR(120), PLANHR(120), LOTINS(120), AQDR(120),	11330002
1BQDR(120), CQDR(120), DQDR(120), EQDR(120), ADMNHR(120), SHIPMT(120),	11340002
2WD(120), INTHR(120), REINHR(120), VISIT(120), ECP(120), MTGHR(120),	11350002
3MRB(120), PCO(120), CAO(120), CONTR(120), ISTRAT(120),	11360099
4DLRIN(120), DLROUT(120), ACNTRT(120), BCNTRT(120), OCNTRT(120),	11370002
5QALIIN(120), QALIRE(120), AONHND(120), BONHND(120), OONHND(120),	11380002
6DLROH(120), NQDR(120), PVINP(120), PEELNP(120), NQAR(120),	11390099
7ISEQ(120), RECS(120), DAYSCL(120), PEHR(120), PVIHR(120), IPRNT(120),	11400044
8IPEER(120)	11410098
REAL WMDR(120), SI(120), EPA(120), DEVN(120)	11420099
COMMON /CHTR/BFSCM, FLAG	11430002
IF (ICIP .EQ. 1) GOTO 1	11440002
CALL CIP(J, FSCM, JCIP, ICIPNO, TYP)	11450098
1 DCASR(1)=DCASR(J)	11460002
QAORG(1)=QAORG(J)	11470002
FSCM(1)=FSCM(J)	11480002
TYP(1)=TYP(J)	11490002
PVN(1)=PVN(J)	11500002
CMDTY(1)=CMDTY(J)	11510002
MONTH(1)=MONTH(J)	11520002
YEAR(1)=YEAR(J)	11530002
PLANHR(1)=PLANHR(J)	11540002
PVIHR(1)=PVIHR(J)	11550002
LOTINS(1)=LOTINS(J)	11560002
PEHR(1)=PEHR(J)	11570002
AQDR(1)=AQDR(J)	11580002
BQDR(1)=BQDR(J)	11590002
CQDR(1)=CQDR(J)	11600002
DQDR(1)=DQDR(J)	11610002
EQDR(1)=EQDR(J)	11620002
ADMNHR(1)=ADMNHR(J)	11630002
SHIPMT(1)=SHIPMT(J)	11640002
WD(1)=WD(J)	11650002
INTHR(1)=INTHR(J)	11660002
REINHR(1)=REINHR(J)	11670002
VISIT(1)=VISIT(J)	11680002
ECP(1)=ECP(J)	11690002
MTGHR(1)=MTGHR(J)	11700002
MRB(1)=MRB(J)	11710002
PCO(1)=PCO(J)	11720002
CAO(1)=CAO(J)	11730002
SI(1)=SI(J)	11740002
EPA(1)=EPA(J)	11750092
DEVN(1)=DEVN(J)	11760099
CONTR(1)=CONTR(J)	11770002
DLRIN(1)=DLRIN(J)	11780002
DLROUT(1)=DLROUT(J)	11790002
ACNTRT(1)=ACNTRT(J)	11800002
BCNTRT(1)=BCNTRT(J)	11810002
OCNTRT(1)=OCNTRT(J)	11820002
QALIIN(1)=QALIIN(J)	11830002
QALIRE(1)=QALIRE(J)	11840002
AONHND(1)=AONHND(J)	11850002
BONHND(1)=BONHND(J)	11860002
OONHND(1)=OONHND(J)	11870002
DLROH(1)=DLROH(J)	11880002
NQDR(1)=NQDR(J)	11890002
PVINP(1)=PVINP(J)	11900002
PEELNP(1)=PEELNP(J)	11910002
QAR(1)=QAR(J)	11920034
NQAR(1)=NQAR(J)	11930002
OPER(1)=OPER(J)	11940002
ISTRAT(1)=ISTRAT(J)	11950002
IPEER(1)=IPEER(J)	11960053
WMDR(1)=WMDR(J)	11970002
DAYSCL(1)=DAYSCL(J)	11980002
ISEQ(1)=ISEQ(J)	11990002
RECS(1)=RECS(J)	12000002
IPRNT(1)=IPRNT(J)	12020042
RETURN	12030002

```

C      END
C      *****
C      THIS SUBROUTINE COMPUTES ATTRIBUTE VALUES AND RATES PRIOR TO
C      TOPSIS PROCESSING.
C
C      SUBROUTINE PREPIN(KA,AQDR,BQDR,CQDR,DQDR,EQDR,WD,ECP,MRB,EPA,DEVN,
1WMDR,TOPCA,CARATE,WDRATE,ECPRAT,AMRBRA,EPARAT,DEVNRA,WMDRRA,
2STARTM,STARTY,ENDMO,ENDYR,FSCM,MONTH,YEAR,TOPWD,TOPECP,TPMRB,
3TOPEPA,TOPDEV,TPMDR,DCASR,QAORG,DEGREE,AIDEAL,AWORST,BIDEAL,
4BWORST,ISTRAT,WEIGHT,REINHR,INTHR,MTGHR,SHIPMT,LOTINS,VISIT,PVN,
5TYP,PEHR,CONTR,PLANHR,DAYSCL,PCQ,CAQ,NQDR,SI,ADMNHR,PVINP,
6PEELNP,QALIIN,QALIRE,LONG,TOPSCR,PVIHR,ONHAND,GAR,IPRNT,
7IPEER,CMDTY,NQAR)
C      INTEGER AQDR(120),BQDR(120),CQDR(120),DQDR(120),EQDR(120),WD(120),
1ECP(120),MRB(120),STARTM,STARTY,ENDMO,ENDYR,MONTH(120),YEAR(120),
2IMONTH(120),IYEAR(120),DEGREE(120),IDGREE(120),ISTRAT(120),
3JSTRAT(120),REINHR(120),INTHR(120),MTGHR(120),SHIPMT(120),
4LOTINS(120),VISIT(120),PEHR(120),CONTR(120),PLANHR(120),
5DAYSCL(120),PCQ(120),CAQ(120),NQDR(120),ADMNHR(120),PVINP(120),
6PEELNP(120),QALIIN(120),QALIRE(120),FLAGA(120)
C      INTEGER FLAGB(120),FLAGC1(120),FLAGC2(120),FLAGC3(120),FLAGC4(120)
1FLAGC5(120),FLAGD(120),FLAGEF(120),FLAGE(120),FLAGF(120),
2FLAGG2(120),FLAGH1(120),FLAGH2(120),FLAGJ(120),FLAGK1(120)
C      INTEGER FLAGK2(120),FLAGL(120),FLAGN(120),FLAGO(120),FLAGP(120),
1FLAGQ1(120),FLAGQ3(120),FCOUNT(120),FLAGE1(120),
2PVIHR(120),ONHAND(120),IPRNT(120),IPEER(120),JPEER(120),NQAR(120)
C      CHARACTER FSCM(120)*6,DCASR(120)*6,QAORG(120)*3,DCASCD(120)*6,
1ORGCD(120)*3,PVN(120)*1,TYP(120)*1,FLAGF1(120)*1,FLAGG1(120)*1,
2FLAG(120)*18,TP(120)*1,QAR(120)*5,CMDTY(120)*2,BFSCM(2000)*5
C      REAL WMDR(120),SI(120),EPA(120),DEVN(120),TOPCA(120),CARATE(120),
1WDRATE(120),ECPRAT(120),AMRBRA(120),EPARAT(120),DEVNRA(120),
2WMDRRA(120),TOPEPA(120),TOPDEV(120),TPMRB(120),TOPECP(120),
3TOPWD(120),TPMDR(120),A(999,38),TOPSCR(14,120),AIDEAL(4,7),
4AWORST(4,7),BIDEAL(4,7),BWORST(4,7),WEIGHT(7),FLAGM(120)
C      COMMON /RVAR/A
C      COMMON /CHTR/BFSCM,FLAG
C      INITIALIZE ARRAYS. ATTRIBUTE DEFAULT VALUES ARE O.O. RATE DEFAULT
C      VALUES ARE O.O.
C      DO 5 I=1,120
C      TOPCA(I)=O.O
C      CARATE(I)=O.O
C      TOPEPA(I)=O.O
C      EPARAT(I)=O.O
C      TOPDEV(I)=O.O
C      DEVNRA(I)=O.O
C      TPMRB(I)=O.O
C      AMRBRA(I)=O.O
C      TOPECP(I)=O.O
C      ECPRAT(I)=O.O
C      TOPWD(I)=O.O
C      WDRATE(I)=O.O
C      TPMDR(I)=O.O
C      WMDRRA(I)=O.O
C      5 CONTINUE
C      COMPUTE THE NUMBER OF MONTHS TOPSIS WILL PROCESS.
C      MON=(ENDYR-STARTY)*12 + ENDMO - STARTM + 1
C      IF((YEAR(KA) LT. ENYR) .OR. ((YEAR(KA) .EQ. ENYR) .AND.
1(MONTH(KA) LT. ENDMO))) MON=MON-((ENDYR-YEAR(KA))*12+ENDMO-
2MONTH(KA))
C      IF (MON .GE. KA) GOTO 30
C      DO 10 I=1,MON
C      ***** COMPUTE RATES FOR ATTRIBUTES.RATES RANGE FROM -3 TO +3.
C      RATE IS COMPUTED BY TAKING RATIO OF FIRST HALF TO SECOND HALF
C      OF DATA ARRAY. RATES OF 0 TO 3 ARE INCREASES.
C      INITIALIZE INTERNAL PARAMETERS
C      IPRICA=O
C      IAFTCA=O
C      PRIEPA=O.O
C      AFTEPA=O.O
C      IPRIMR=O
C      IAFTMR=O
C      IPRPWD=O
C      IAFTWD=O
C      PRIDEV=O.O
C      AFTDEV=O.O
C      IPRIEC=O
C      IAFTEC=O
C      PRIMDR=O.O
C      AFTMDR=O.O
C      IF THERE IS MORE THAN 6 MONTHS OF DATA, MODEL IGNORES PRIOR

```


C	DATA WHEN COMPUTING RATES.	12840002
	JMON=MON-1	12850002
	KC=KA-JMON	12850002
	IF(KC .GT. 6) GOTO 7	12870045
C	IF(KC .GT. 12) GOTO 7	12880045
	IOFFST=1	12890002
	GO TO 6	12900002
C	7 IOFFST=KC-11	12910045
C	KC=12	12920045
	7 IOFFST=KC-5	12930045
	KC=6	12940045
	6 MIDMON=(KC + 1)/2	12950002
	IF (MIDMON*2 .EQ. (KC + 1)) GOTO 22	12960002
C	EVEN NUMBER OF DATA POINTS	12970002
	INDMO=MIDMON+IOFFST-1	12980002
	GO TO 21	12990002
C	ODD NUMBER OF DATA POINTS	13000002
	22 INDMO=MIDMON+IOFFST-2	13010002
	21 DO 24 J=IOFFST,INDMO	13020002
	K=MIDMON + J	13030002
C	THE NEXT TWO LINES REFLECT POLICY CHANGE UNDER IQUE	13040099
C	CARS ARE REPORTED AS COMBINED VERBAL AND WRITTEN VS TYPE A + B	13050099
C	MUST BE CAREFUL HERE .	13060099
	IPRICA=IPRICA+((AQDR(J)+BQDR(J))*7.5+CQDR(J)*30+DQDR(J)*60	13070099
	1+EQDR(J)*30)	13080002
	IAFTCA=IAFTCA+((AQDR(K)+BQDR(K))*7.5+CQDR(K)*30+DQDR(K)*60	13090099
	1+EQDR(K)*30)	13100002
	PRIEPA=PRIEPA+EPA(J)	13110002
	AFTEPA=AFTEPA+EPA(K)	13120002
	IPRMR=IPRMR+MRB(J)	13130002
	IAFTMR=IAFTMR+MRB(K)	13140002
	IPRIWD=IPRIWD+WD(J)	13150002
	IAFTWD=IAFTWD+WD(K)	13160002
	PRIDEV=PRIDEV+DEVN(J)	13170099
	AFTDEV=AFTDEV+DEVN(K)	13180099
	IPRIEC=IPRIEC+ECP(J)	13190002
	IAFTEC=IAFTEC+ECP(K)	13200002
	PRIMDR=PRIMDR+WMDR(J)	13210002
	AFTMDR=AFTMDR+WMDR(K)	13220002
	24 CONTINUE	13230002
	L=KA-MON+1	13240002
C	**** COMPUTE ATTRIBUTE VALUES.	13250002
	TOPCA(I)=(AQDR(L)+BQDR(L))*7.5+CQDR(L)*30+DQDR(L)*60+EQDR(L)*30	13260099
	TOPEPA(I)=EPA(L)	13270002
	TOPMRB(I)=MRB(L)	13280002
	TOPWD(I)=WD(L)	13290002
	TOPDEV(I)=DEVN(L)	13300099
	TOPECP(I)=ECP(L)	13310002
	TOPMDR(I)=WMDR(L)	13320002
C	RATE IS A VALUE BETWEEN -3 AND +3	13330099
C	IF SECOND HALF IS NEGLIGIBLE OR ZERO, RATE IS -3 OR 0 DEPENDING ON	13340002
C	FIRST HALF ACTIVITY.	13350002
C	RATE IS SET TO -3.0 WHEN THERE IS NO HISTORY WHATSOEVER OF INDICATOR.	13360023
C	THIS WAS DECIDED AT SAG #5 AND DOCUMENTED VIA MFR 22NOV 88.	13370023
C		13380023
	26 CARATE(I)=(REAL(IPRICA))/(REAL(IAFTCA)+.1)*3.0	13390002
	IF(CARATE(I) .GT. 6.0) CARATE(I)=6.0	13400002
	IF((IPRICA+IAFTCA) .EQ. 0) CARATE(I)=6.0	13410023
	CARATE(I)=3.0-CARATE(I)	13420002
	EPARAT(I)=PRIEPA/(AFTEPA+.01)*3.0	13430002
	IF(EPARAT(I) .GT. 6.0) EPARAT(I)=6.0	13440002
	IF((PRIEPA+AFTEPA) .EQ. 0.0) EPARAT(I)=6.0	13450023
	EPARAT(I)=3.0-EPARAT(I)	13460002
	AMRBRA(I)=(REAL(IPRMR))/(REAL(IAFTMR)+.01)*3.0	13470002
	IF(AMRBRA(I) .GT. 6.0) AMRBRA(I)=6.0	13480002
	IF((IPRMR+IAFTMR) .EQ. 0) AMRBRA(I)=6.0	13490023
	AMRBRA(I)=3.0-AMRBRA(I)	13500002
	WDRATE(I)=(REAL(IPRIWD))/(REAL(IAFTWD)+.01)*3.0	13510002
	IF(WDRATE(I) .GT. 6.0) WDRATE(I)=6.0	13520002
	IF((IPRIWD+IAFTWD) .EQ. 0) WDRATE(I)=6.0	13530023
	WDRATE(I)=3.0-WDRATE(I)	13540002
	DEVNRA(I)=PRIDEV/(AFTDEV+.01)*3.0	13550099
	IF(DEVNRA(I) .GT. 6.0) DEVNRA(I)=6.0	13560099
	IF((PRIDEV+AFTDEV) .EQ. 0.0) DEVNRA(I)=6.0	13570099
	DEVNRA(I)=3.0-DEVNRA(I)	13580099
	ECPRAT(I)=(REAL(IPRIEC))/(REAL(IAFTEC)+.01)*3.0	13590002
	IF(ECPRAT(I) .GT. 6.0) ECPRAT(I)=6.0	13600002
	IF((IPRIEC+IAFTEC) .EQ. 0) ECPRAT(I)=6.0	13610023
	ECPRAT(I)=3.0-ECPRAT(I)	13620002
	WMDRRA(I)=PRIMDR/(AFTMDR+.01)*3.0	13630002

```

      IF(WMDRRA(I) .GT. 6.0) WMDRRA(I)=6.0          13640002
      IF((PRIMDR+AFTMDR) .EQ. 0.0) WMDRRA(I)=6.0    13650023
      WMDRRA(I)=3.0-WMDRRA(I)                        13660002
C SHIFT OTHER VARIABLES TO THE NEW TIME FRAME       13670002
      CALL SHIFTR(I,L,DCASCD,DCASR,ORGCD,QAORG,IMONTH,MONTH,IYEAR,YEAR, 13680002
      1IDGREE,DEGREE,JSTRAT,ISTRAT,FLAGA,REINHR,FLAGB,INTHR,FLAGD,MTGHR, 13690002
      2FLAGEF,SHIPMT,FLAGE,LOTINS,FLAGF,VISIT,FLAGF1,TYP,TP,FLAGG1,PVN, 13700002
      3FLAGG2,PEHR,FLAGH1,CONTR,FLAGH2,PLANHR,FLAGJ,DAYSCL,FLAGK1,OALIIN, 13710002
      4FLAGK2,QALIRE,FLAGL,NQDR,FLAGM,SI,FLAGN,PCO,FLAGO,CAO,FLAGP,      13720092
      5ADMNHR,FLAGQ1,PVINP,FLAGQ3,PEELNP,FLAGC1,AQDR,                    13730099
      6FLAGC2,BQDR,FLAGC3,CQDR,FLAGC4,DQDR,FLAGC5,EQDR,FLAGE1,PVIHR,    13740026
      7ONHAND,QAR,IPRNT,IPEER,JPEER,CMDTY,NQAR)      13750099
C COMPUTE TOPSIS SCORES                            13760002
      CALL TOPSIS(I,IDGREE,TOPCA,CARATE,TOPEPA,EPARAT,TOPMRB,AMRBRA,    13770002
      1TOPWD,WDRATE,TOPDEV,DEVNRA,TOPECP,ECPRAT,TOPMDR,WMDRRA,TOPSCR,    13780099
      2AIDEAL,AWORST,BIDEAL,BWORST,JPEER,WEIGHT,LOTINS)                13790092
C IDENTIFY RED FLAG CONDITIONS                     13800002
      CALL FLGGR(I,FLAGA,TOPCA,FLAGB,FLAGC1,FLAGC2,FLAGC3,FLAGC4,      13810002
      1FLAGC5,FLAGD,FLAGEF,FLAGE,FLAGE1,                                13820002
      2FLAGF,FLAGF1,FLAGG1,FLAGG2,FLAGH1,FLAGH2,TOPDEV,                13830099
      3FLAGJ,FLAGK1,FLAGK2,ONHAND,FLAGL,FLAGM,FLAGN,FLAGO,FLAGP,      13840026
      4FLAGQ1,FLAGQ2,FCOUNT,JPEER,TOPEPA)                    13850099
      CALL SCORER(I,FCOUNT,TOPSCR,TP,JSTRAT)                13860002
C WRITE RECORD TO VERIFY PROGRAM                    13870002
      IF (LONG .EQ. 0) GOTO 50                                     13880002
      WRITE(9,25) I,DCASCD(I),ORGCD(I),FSCM(I),IMONTH(I),IYEAR(I),    13890002
      1JPEER(I),IDGREE(I),TOPEPA(I),EPARAT(I),TOPDEV(I),DEVNRA(I),      13900099
      2TOPMRB(I),AMRBRA(I),TOPWD(I),WDRATE(I),TOPECP(I),              13910060
      3ECPRAT(I),TOPCA(I),CARATE(I),TOPMDR(I),WMDRRA(I),              13920060
      4TOPSCR(1,I),TOPSCR(2,I),TOPSCR(3,I),TOPSCR(4,I),TOPSCR(5,I),    13930002
      5TOPSCR(6,I),TOPSCR(7,I),TOPSCR(8,I),TOPSCR(9,I),TOPSCR(10,I),   13940002
      6TOPSCR(11,I),FLAG(I),FCOUNT(I)                            13950002
25  FORMAT(13,A6,A3,A6,2I3,I4,I2,2(F6.2,F5.2),4(F4.0,F5.2),F4.2,F5.2, 13960099
      11,3,1,A18,I2)                                             13970099
50  IF((IYEAR(I) .LT. STARTY) .OR. ((IYEAR(I) .EQ. STARTY) .AND.      13980002
      1(IMONTH(I) .LT. STARTM))) GOTO 10                          13990002
      WRITE(11,51) ORGCD(I),FSCM(I),TP(I),CMDTY(I),PVN(I),NQAR(I),    14000099
      1IMONTH(I),                                                14010099
      2IYEAR(I),JSTRAT(I),JPEER(I),IDGREE(I),QAR(I),FLAG(I),TOPSCR(10,I), 14020099
      3TOPSCR(1,I),TOPSCR(2,I),TOPSCR(3,I),TOPSCR(4,I),TOPSCR(5,I),    14030099
      4TOPSCR(6,I),TOPSCR(7,I),TOPSCR(8,I),TOPSCR(11,I),TOPSCR(12,I),  14040099
      5TOPSCR(13,I),TOPSCR(14,I),STARTM,STARTY,ENDMO,ENDYR,I        14050099
51  FORMAT(A3,A6,A1,A2,A1,3I3,2I4,I2,A5,A18,13F7.1,5I2)        14060099
10  CONTINUE                                                  14070002
      GO TO 40                                                  14080002
30  WRITE (6,35) FSCM(I),KA,MON                                14090002
35  FORMAT(2X,'WARNING. INSUFFICIENT DATA FOR FSCM',A6,2I3,        14100002
      1'FSCM SKIPPED BUT DATA ON THE LABEL FILE')              14110002
40  RETURN                                                    14120002
      END                                                        14130002
      SUBROUTINE TOPSIS(I,IDGREE,TOPCA,CARATE,TOPEPA,EPARAT,TOPMRB,    14140002
      1AMRBRA,TOPWD,WDRATE,TOPDEV,DEVNRA,TOPECP,ECPRAT,TOPMDR,WMDRRA,  14150099
      2TOPSCR,AIDEAL,AWORST,BIDEAL,BWORST,JPEER,WEIGHT,LOTINS)        14160092
      REAL A(999,38),TOPCA(120),CARATE(120),TOPEPA(120),EPARAT(120),  14170052
      1TOPMRB(120),AMRBRA(120),TOPWD(120),WDRATE(120),TOPDEV(120),    14180099
      2DEVNRA(120),TOPECP(120),ECPRAT(120),TOPMDR(120),WMDRRA(120),  14190099
      3TOPSCR(14,120),AIDEAL(4,7),AWORST(4,7),BIDEAL(4,7),BWORST(4,7), 14200099
      4Z(7),SPLUS(7),SMINUS(7),WEIGHT(7)                        14210092
      INTEGER IDGREE(120),JPEER(120),LOTINS(120)                14220060
      COMMON /RVAR/A                                              14230002
      TOPSCR(8,I)=0.0                                             14240002
      VTSUM=0.0                                                  14250002
      SPOS=0.0                                                  14260002
      SNEG=0.0                                                  14270002
C COMPUTE Z VALUES FOR NONRATE PARAMETERS          14280002
C NEXT LINES ASSUME EXPONENTIAL DISTRIBUTION        14290002
      Z(1)=LOG(TOPEPA(I)/(A(JPEER(I),13)+.0001)+.00001)          14300060
      Z(2)=LOG(TOPDEV(I)/(A(JPEER(I),15)+.0001)+.00001)          14310099
      Z(3)=LOG(TOPMRB(I)/(A(JPEER(I),7)+.0001)+.00001)           14320060
      Z(4)=LOG(TOPWD(I)/(A(JPEER(I),3)+.0001)+.00001)            14330060
      Z(5)=LOG(TOPECP(I)/(A(JPEER(I),5)+.0001)+.00001)           14340060
      Z(6)=LOG(TOPCA(I)/(A(JPEER(I),29)+.0001)+.00001)           14350060
      Z(7)=LOG(TOPMDR(I)/(A(JPEER(I),35)+.0001)+.00001)          14360060
C ASSIGN LIMITS TO Z VALUES FOR OUTLIERS.Z MUST BE BETWEEN -3.0 AND 3.0 14370032
      DO 1 J=1,7                                                  14380002
      IF(Z(J) .LT. -3.0) Z(J)=-3.0                                14390002
      IF(Z(J) .GT. 3.0) Z(J)=3.0                                  14400002
1    CONTINUE                                                  14410002
C COMPUTE TOPSIS SEPARATION MEASURES FROM NEGATIVE IDEAL 14420002
      SMINUS(1)=BWORST(IDGREE(I),1)-EPARAT(I)                  14430002

```

```

SMINUS(2)=BWorst(IDGREE(1),2)-DEVNRA(1) 14440099
SMINUS(3)=BWorst(IDGREE(1),3)-AMRBRA(1) 14450002
SMINUS(4)=BWorst(IDGREE(1),4)-WDRATE(1) 14460002
SMINUS(5)=BWorst(IDGREE(1),5)-ECPRAT(1) 14470002
SMINUS(6)=BWorst(IDGREE(1),6)-CARATE(1) 14480002
SMINUS(7)=BWorst(IDGREE(1),7)-WMDRRA(1) 14490002
C IF CONTRACTOR IS A PROBLEM, INCREASE WEIGHT OF TREND TO 60/40 EXC PQDR 14500099
IF((IDGREE(1) .EQ. 1) .OR. (IDGREE(1) .EQ. 3)) THEN 14510002
DO 3 J=1,6 14520099
SMINUS(J)=SMINUS(J)*1.5 14530099
3 CONTINUE 14540099
SMINUS(7)=AWorst(IDGREE(1),7)-Z(7) 14550002
ENDIF 14560099
DO 2 J=1,7 14570002
C SMINUS(J)=SQRT(SMINUS(J)**2+(AWorst(IDGREE(1),J)-Z(J))**2) 14580033
C BELOW LINE REFLECTS 'CITY BLOCK' DISTANCE. ABOVE IS EUCLIDIAN 14590002
C CITY BLOCK SEEMS TO WORK BETTER FOR NONRESIDENT. 14600002
SMINUS(J)=SMINUS(J)+(AWorst(IDGREE(1),J)-Z(J)) 14610033
2 CONTINUE 14620002
C COMPUTE TOPSIS SEPARATION MEASURES FROM POSITIVE IDEAL 14630002
SPLUS(1)=EPARAT(1)-BIDEAL(IDGREE(1),1) 14640002
SPLUS(2)=DEVNRA(1)-BIDEAL(IDGREE(1),2) 14650099
SPLUS(3)=AMRBRA(1)-BIDEAL(IDGREE(1),3) 14660002
SPLUS(4)=WDRATE(1)-BIDEAL(IDGREE(1),4) 14670002
SPLUS(5)=ECPRAT(1)-BIDEAL(IDGREE(1),5) 14680002
SPLUS(6)=CARATE(1)-BIDEAL(IDGREE(1),6) 14690002
SPLUS(7)=WMDRRA(1)-BIDEAL(IDGREE(1),7) 14700002
IF((IDGREE(1) .EQ. 1) .OR. (IDGREE(1) .EQ. 3)) THEN 14710002
DO 4 J=1,6 14720099
SPLUS(J)=SPLUS(J)*1.5 14730099
4 CONTINUE 14740099
SPLUS(7)=Z(7)-AIDEAL(IDGREE(1),7) 14750002
ENDIF 14760099
DO 7 J=1,7 14770002
C SPLUS(J)=SQRT(SPLUS(J)**2+(Z(J)-AIDEAL(IDGREE(1),J))**2) 14780033
C BELOW LINE REFLECTS 'CITY BLOCK' DISTANCE. ABOVE IS EUCLIDIAN 14790002
C CITY BLOCK SEEMS TO WORK BETTER FOR NONRESIDENT. 14800002
SPLUS(J)=SPLUS(J)+(Z(J)-AIDEAL(IDGREE(1),J)) 14810033
7 CONTINUE 14820002
C COMBINE DISTANCES INTO AN OVERALL TOPSIS SCORE 14830002
DO 9 J=1,7 14840002
WTSUM=WTSUM+WEIGHT(J) 14850086
SNEG=SNEG+(SMINUS(J)*WEIGHT(J))**2 14860086
SPOS=SPOS+(SPLUS(J)*WEIGHT(J))**2 14870086
TOPSCR(J,1)=SMINUS(J)/(SPLUS(J)+SMINUS(J)+.00001)*100 14880086
TOPSCR(8,1)=TOPSCR(8,1)+TOPSCR(J,1)*WEIGHT(J) 14890086
9 CONTINUE 14900002
TOPSCR(9,1)=SQRT(SNEG)/(SQRT(SNEG)+SQRT(SPOS))*100 14910002
TOPSCR(8,1)=TOPSCR(8,1)/WTSUM 14920002
C COMPUTE A MODIFIED SCORE BASED ON DEGREE OF DIFFICULTY 14930002
CALL LIMITR(1,IDGREE,TOPSCR) 14940023
RETURN 14950002
END 14960002
C 14970002
C *** DEFINES IDEAL AND NEGATIVE IDEAL CONDITIONS FOR EACH DIFFICULTY 14980002
C ALSO DETERMINES WEIGHT FACTORS FOR EACH SITUATION 14990002
C 15000002
SUBROUTINE CORNER(AIDEAL,AWorst,BIDEAL,BWorst,WEIGHT) 15010002
REAL AIDEAL(4,7),AWorst(4,7),BIDEAL(4,7),BWorst(4,7),WEIGHT(7) 15020022
C DEFINE IDEAL AND NEGATIVE IDEALS 15030002
C J=1 - PA, J=2 - DEVN, J=3 - MRB, J=4 - WVR, J=5 - ECP, J=6 - CAR, J=7 - PQDR 15040099
C K IS DIFFICULTY INDEX 15050002
C THESE ARE QUEST III WEIGHTS FROM 17-19 JUL 90 SAG. 15060099
C WEIGHTS MUST ADD TO 1.00 15070099
WEIGHT(1)=.168 15080099
WEIGHT(2)=.074 15090099
WEIGHT(3)=.148 15100099
WEIGHT(4)=.151 15110099
WEIGHT(5)=.064 15120099
WEIGHT(6)=.198 15130099
WEIGHT(7)=.197 15140099
DO 1 K=1,4 15150002
DO 2 J=1,7 15160002
AIDEAL(K,J)=-3.0 15170002
AWorst(K,J)=3.0 15180002
BIDEAL(K,J)=-3.0 15190002
BWorst(K,J)=3.0 15200002
2 CONTINUE 15210002
IF((K .EQ. 1) .OR. (K .EQ. 3)) THEN 15220002
BIDEAL(K,7)=0.0 15230002

```

BWORST(K,7)=0.0	15240002
ENDIF	15250002
C DO 5 J=1,7	15260099
C WRITE(6,10) K,J,AIDEAL(K,J),AWORST(K,J),BIDEAL(K,J),BWORST(K,J)	15270099
C 10 FORMAT(2I2,5F10.2)	15280099
C 5 CONTINUE	15290099
1 CONTINUE	15300099
WRITE(6,12) WEIGHT(1),WEIGHT(2),WEIGHT(3),WEIGHT(4),	15310099
1WEIGHT(5),WEIGHT(6),WEIGHT(7)	15320099
12 FORMAT(/,7F10.3)	15330099
RETURN	15340002
END	15350002
C	15360002
C THIS SUBROUTINE ESTABLISHES ARBITRARY LIMITS ON TOPSIS SCORES	15370002
C A KNOB FOR PROBLEM FACILITIES IS SET TO MAX PRODUCT SCORES AT 79%	15380023
C A KNOB FOR PROBLEM FACILITIES IS REMOVED.	15390099
C	15400023
SUBROUTINE LIMITR(I,IDGREE,TOPSCR)	15410023
REAL TOPSCR(14,120)	15420099
INTEGER IDGREE(120)	15430002
COMMON /RVAR/A	15440002
DO 2 J=8,9	15450002
IF ((IDGREE(I) .EQ. 1) .OR. (IDGREE(I) .EQ. 3)) TOPSCR(J,I)=	15460002
1TOPSCR(J,I)*1.00	15470099
2 CONTINUE	15480002
RETURN	15490023
END	15500002
C	15510002
C THIS SUBROUTINE LINKS HISTORY ARRAYS WITH MODEL ARRAYS	15520002
C	15530002
SUBROUTINE SHIFTR(I,L,DCASCD,DCASR,ORGCD,QAORG,IMNTH,MNTH,IYR,YR,	15540002
1IDGREE,DEGREE,JSTRAT,ISTRAT,FLAGA,REINHR,FLAGB,INTHR,FLAGD,MTGHR,	15550002
2FLAGEF,SHIPMT,FLAGE,LOTINS,FLAGF,VISIT,FLAGF1,TYP,TP,FLAGG1,PVN,	15560002
3FLAGG2,PEHR,FLAGH1,CONTR,FLAGH2,PLANHR,FLAGJ,DAYSCL,FLAGK1,QALIIN,	15570002
4FLAGK2,QALIRE,FLAGL,NQDR,FLAGM,SI,FLAGN,PCO,FLAGO,CAO,FLAGP,	15580092
5ADMNHR,FLAGQ1,PVINP,FLAGQ3,PEELNP,FLAGC1,AQDR,	15590099
6FLAGC2,BQDR,FLAGC3,CQDR,FLAGC4,DQDR,FLAGC5,EQDR,FLAGE1,PVIHR,	15600026
7ONHAND,QAR,IPRNT,IPEER,JPEER,CMDTY,NQAR)	15610099
INTEGER MNTH(120),YR(120),AQDR(120),BQDR(120),CQDR(120),DQDR(120),	15620002
1IMNTH(120),IYR(120),DEGREE(120),IDGREE(120),ISTRAT(120),	15630002
2JSTRAT(120),REINHR(120),INTHR(120),MTGHR(120),SHIPMT(120),	15640002
3LOTINS(120),VISIT(120),PEHR(120),CONTR(120),PLANHR(120),	15650002
4DAYSCL(120),PCO(120),CAO(120),NQDR(120),ADMNHR(120),PVINP(120),	15660002
5PEELNP(120),QALIIN(120),QALIRE(120),FLAGA(120),	15670099
6FLAGB(120),FLAGD(120),FLAGEF(120),FLAGE(120),FLAGF(120),	15680002
7FLAGG2(120),FLAGH1(120),FLAGH2(120),FLAGJ(120),FLAGK1(120),	15690002
8FLAGK2(120),FLAGL(120),FLAGN(120),FLAGO(120),FLAGP(120),	15700002
9FLAGQ1(120),FLAGQ3(120),EQDR(12),FLAGC1(120),NQAR(120)	15710099
INTEGER FLAGC2(120),FLAGC3(120),FLAGC4(120),FLAGC5(120),	15720002
1FLAGE1(120),PVIHR(120),ONHAND(120),IPRNT(120),IPEER(120),	15730061
2JPEER(120)	15740061
CHARACTER DCASR(120)*6,QAORG(120)*3,DCASCD(120)*6,TP(120)*1,	15750002
1ORGCD(120)*3,PVN(120)*1,TYP(120)*1,FLAGF1(120)*1,FLAGG1(120)*1,	15760034
2QAR(120)*5,CMDTY(120)*2	15770099
REAL SI(120),FLAGM(120)	15780092
DCASCD(I)=DCASR(L)	15790002
CMDTY(I)=CMDTY(L)	15800095
PVN(I)=PVN(L)	15810095
ORGCD(I)=QAORG(L)	15820002
IMNTH(I)=MNTH(L)	15830002
IYR(I)=YR(L)	15840002
IDGREE(I)=DEGREE(L)	15850002
QAR(I)=QAR(L)	15860034
NQAR(I)=NQAR(L)	15870099
JSTRAT(I)=ISTRAT(L)	15880002
JPEER(I)=IPEER(L)	15890060
TP(I)=TYP(L)	15900002
IPRNT(I)=IPRNT(L)	15910042
FLAGA(I)=REINHR(L)	15920002
FLAGB(I)=INTHR(L)	15930002
FLAGC1(I)=AQDR(L)	15940002
FLAGC2(I)=BQDR(L)	15950002
FLAGC3(I)=CQDR(L)	15960002
FLAGC4(I)=DQDR(L)	15970002
FLAGC5(I)=EQDR(L)	15980002
FLAGD =MTGHR(L)	15990002
FLAGEF(I)=SHIPMT(L)	16000002
FLAGE(I)=LOTINS(L)	16010002
FLAGE1(I)=PVIHR(L)	16020002
FLAGF(I)=VISIT(L)	16030002

```

FLAGF1(I)=TYP(L)
FLAGG1(I)=PVN(L)
FLAGG2(I)=PEHR(L)
FLAGH1(I)=CONTR(L)
FLAGH2(I)=PLANHR(L)
FLAGJ(I)=DAYSCL(L)
FLAGK1(I)=QALIIIN(L)
FLAGK2(I)=QALIRE(L)
ONHAND(I)=ONHAND(L)
FLAGL(I)=NQDR(L)
FLAGM(I)=SI(L)
FLAGN(I)=PCO(L)
FLAGO(I)=CAO(L)
FLAGP(I)=ADMNHR(L)
FLAGQ1(I)=PVINP(L)
FLAGQ3(I)=PEELNP(L)
RETURN
END
C *** THIS SUBROUTINE CHECKS FOR OUT OF TOLERANCE CONDITIONS***
SUBROUTINE FLGGR(I,FLAGA,TOPCA,FLAGB,FLAGC1,FLAGC2,FLAGC3,
1FLAGC4,FLAGC5,FLAGD,FLAGEF,FLAGE,FLAGE1,FLAGF,FLAGF1,FLAGG1,
2FLAGG2,FLAGH1,FLAGH2,TOPDEV,FLAGJ,FLAGK1,FLAGK2,ONHAND,FLAGL,
3FLAGM,FLAGN,FLAGO,FLAGP,FLAGQ1,FLAGQ3,FCOUNT,JPEER,TOPEPA)
INTEGER FLAGA(120),FLAGB(120),FLAGC1(120),FLAGC2(120),FLAGC3(120),
1FLAGC4(120),FLAGD(120),FLAGEF(120),FLAGE(120),FLAGF(120),
2FLAGG2(120),FLAGH1(120),FLAGH2(120),FLAGJ(120),FLAGK1(120),
3FLAGK2(120),FLAGL(120),FLAGN(120),FLAGO(120),FLAGP(120),
4FLAGQ1(120),FLAGQ3(120),FLAGC5(120),FCOUNT(120),
5JPEER(120),FLAGE1(120),ONHAND(120)
REAL A(999,38),TOPCA(120),TOPDEV(120),FLAGM(120),TOPEPA(120)
CHARACTER FLAGG1(120)*1,FLAGF1(120)*1,FLAG(120)*18,BFSCM(2000)*5
COMMON /RVAR/A
COMMON /CHTR/BFSCM,FLAG
FLAG(I)=' '
FCOUNT(I)=0
C CHECK FOR FLAG CONDITIONS A THRU O
C FLAG C, QDR DISTRIBUTIONS REQUIRE ANOTHER SUBROUTINE
C FLAG A OCCURS WHEN THERE ARE LOTS REJECTED AND NO CORRECTIVE ACTIONS
C OF AT LEAST TYPE B FOR TWO CONSECUTIVE MONTHS.FOR FACILITIES
C THAT HAVE NO LOTS, THE EPA IS USED(UNITS HAVE BEEN REJECTED)
IF (I .EQ. 1) GOTO 14
IF(((FLAGC2(I)+FLAGC3(I)+FLAGC4(I)+FLAGC5(I)) .GT. 0) GOTO 14
IF((TOPDEV(I-1) .GT. 0.0) .AND. ((FLAGC2(I-1)+FLAGC3(I-1)+
1FLAGC4(I-1)+FLAGC5(I-1)) .EQ. 0)) THEN
FLAG(I)(1:1)='A'
FCOUNT(I)=FCOUNT(I)+1
GO TO 14
ELSE
IF(FLAGE(I-1) .GT. 0) GOTO 14
IF(TOPEPA(I-1) .GT. 0.0) THEN
IF((FLAGC2(I-1)+FLAGC3(I-1)+FLAGC4(I-1)+FLAGC5(I-1)) .GT. 0)
1GOTO 14
FLAG(I)(1:1)='A'
FCOUNT(I)=FCOUNT(I)+1
ENDIF
14 ENDIF
C FLAG B OCCURS WHEN MODEL DETECTS INTENSIFIED INSP HOURS BUT NO
C CORRECTIVE ACTION OF AT LEAST TYPE B. MODEL LOOKS BACK ONE
C MONTH TO SEE IF QDR WRITTEN.
IF((FLAGB(I) .GT. 0) .AND. ((FLAGC2(I)+FLAGC3(I)+FLAGC4(I)+
1FLAGC5(I)) .EQ. 0)) THEN
IF(I .EQ. 1) GOTO 3
IF((FLAGC2(I-1)+FLAGC3(I-1)+FLAGC4(I-1)+FLAGC5(I-1)) .GT. 0)
1GOTO 3
FLAG(I)(2:2)='B'
FCOUNT(I)=FCOUNT(I)+1
3 ENDIF
C FLAG C OCCURS WHEN CORRECTIVE ACTION DISTRIBUTION IS ABNORMAL
CALL DISTR(I,FLAGC1,FLAGC2,FLAGC3,FLAGC4,FLAGC5,FCOUNT,JPEER)
C FLAG D OCCURS WHEN MODEL DETECTS THREE MONTHS OF NO MTG HOURS
IF(I .LE. 2) GOTO 4
IF((FLAGD(I-2) .EQ. 0) .AND. (FLAGD(I-1) .EQ. 0) .AND.
1(FLAGD(I) .EQ. 0)) THEN
FLAG(I)(4:4)='D'
FCOUNT(I)=FCOUNT(I)+1
4 ENDIF
C FLAG E OCCURS WHEN MODEL DETECTS A SHIPMENT WITHOUT PRODUCT
C VERIFICATION INSPECTION HOURS.MODEL LOOKS BACK ONE
C MONTH TO SEE IF PVI OCCURED.
IF((FLAGEF(I) .GT. 0) .AND. (FLAGE1(I) .EQ. 0)) THEN

```

IF(I .EQ. 1) GOTO 7	16840010
IF(FLAG1(I-1) .GT. 0) GOTO 7	16850002
6 FLAG(I)(5:5)='E'	16860002
FCOUNT(I)=FCOUNT(I)+1	16870002
7 ENDIF	16880002
C FLAG F OCCURS WHEN MODEL DETECTS SHIPMENTS BUT NO VISITS AT	16890002
C A NONRESIDENT FACILITY.	16900002
IF((FLAGF(I) .GT. 0) .AND. (FLAGF1(I) .EQ. 'N') .AND. (FLAGF(I)	16910002
1 .EQ. 0)) THEN	16920002
FLAG(I)(6:6)='F'	16930002
FCOUNT(I)=FCOUNT(I)+1	16940002
ENDIF	16950002
C FLAG G OCCURS UNDER THE FOLLOWING CIRCUMSTANCES	16960002
C 1. MILO OR MILI FACILITY AND	16970002
C IF A RESIDENT FACILITY, THERE IS NO PROCEDURE EVALUATION DURING	16980063
C ANY MONTH - OR	16990002
C 2. MILO OR MILI FACILITY AND	17000063
C IF A NONRESIDENT FACILITY, THERE IS EITHER NO PROCEDURE EVALUATION	17010063
C DURING ANY TWO CONSECUTIVE MONTHS. - OR	17020063
C 3. IF COMMODITY IS P7 AND MILO OR MILI FACILITY AND	17030063
C IF A NONRESIDENT FACILITY, THERE IS EITHER NO PROCESS EVALUATION	17040063
C DURING ANY THREE CONSECUTIVE MONTHS.	17050063
C G FLAG IS SUPPRESSED FOR NUCLEAR FACILITIES	17060066
C	17070066
IF(FLAGG1(I) .EQ. 'C') GO TO 50	17080063
IF((JPEER(I) .GE. 241) .AND. (JPEER(I) .LT. 265)) GO TO 50	17090066
IF((JPEER(I) .GE. 826) .AND. (JPEER(I) .LT. 850)) GO TO 50	17100066
IF((FLAGF1(I) .EQ. 'R') .AND. (FLAGG2(I) .EQ. 0)) THEN	17110063
FLAG(I)(7:7)='G'	17120002
FCOUNT(I)=FCOUNT(I)+1	17130002
GO TO 50	17140063
ENDIF	17150002
IF((FLAGF1(I) .EQ. 'N') .AND. (FLAGG2(I-1) .EQ. 0) .AND.	17160063
1 (FLAGG2(I) .EQ. 0) .AND. (I .GT. 1)) THEN	17170063
FLAG(I)(7:7)='G'	17180002
FCOUNT(I)=FCOUNT(I)+1	17190002
GO TO 50	17200063
ENDIF	17210002
IF((FLAGF1(I) .EQ. 'N') .AND. (FLAGG2(I-1) .EQ. 0) .AND.	17220063
1 (FLAGG2(I) .EQ. 0) .AND. (I .GT. 2) .AND.	17230063
2 (FLAGG2(I-2) .EQ. 0) .AND. ((JPEER(I) .EQ. 283) .OR.	17240063
3 (JPEER(I) .EQ. 284))) THEN	17250064
FLAG(I)(7:7)='G'	17260002
FCOUNT(I)=FCOUNT(I)+1	17270002
ENDIF	17280002
50 CONTINUE	17290063
C IF((FLAGF1(I) .EQ. 'R') .AND. (FLAGG1(I) .NE. 'C') .AND.	17300063
C 1 (FLAGG2(I) .EQ. 0)) THEN	17310063
C FLAG(I)(7:7)='G'	17320063
C FCOUNT(I)=FCOUNT(I)+1	17330063
C ENDIF	17340063
C IF((FLAGF1(I) .EQ. 'N') .AND. (FLAGG1(I) .NE. 'C') .AND.	17350063
C 1 (FLAGG2(I) .EQ. 0) .AND. (I .EQ. 1)) THEN	17360063
C FLAG(I)(7:7)='G'	17370063
C FCOUNT(I)=FCOUNT(I)+1	17380063
C ENDIF	17390063
C IF((FLAGF1(I) .EQ. 'N') .AND. (FLAGG1(I) .NE. 'C') .AND.	17400063
C 1 (FLAGG2(I) .EQ. 0) .AND. (I .GT. 1)) THEN	17410063
C IF(FLAGG2(I-1) .EQ. 0) THEN	17420063
C FLAG(I)(7:7)='G'	17430063
C FCOUNT(I)=FCOUNT(I)+1	17440063
C ENDIF	17450063
C ENDIF	17460063
C FLAG H OCCURS WHEN MODEL DETECTS A CONTRACT RECEIVED LAST MONTH AND	17470002
C NO PLANNING HOURS IN CURRENT MONTH OR PREVIOUS MONTH.	17480002
IF(I .EQ. 1) GOTO 9	17490002
IF((FLAGH1(I-1) .GT. 0) .AND. (FLAGH2(I-1) .EQ. 0)) THEN	17500002
IF(FLAGH2(I) .GT. 0) GOTO 9	17510002
8 FLAG(I)(8:8)='H'	17520002
FCOUNT(I)=FCOUNT(I)+1	17530002
9 ENDIF	17540002
C FLAG I OCCURS WHEN MODEL DETECTS LOTS REJECTED AND NO REINSPECTION	17550002
C HOURS DURING MONTH OR NEXT MONTH. IF NO LOTS ARE INSPECTED, UNIT	17560010
C REJECTIONS WILL TRIGGER THE FLAG.	17570010
IF(I .EQ. 1) GOTO 11	17580009
IF(((TOPDEV(I-1) .GT. 0.0) .OR. ((TOPEA(I-1) .GT. 0.0) .AND.	17590099
1(FLAG1(I-1) .EQ. 0))) .AND. (FLAGA(I-1) .EQ. 0)) THEN	17600010
IF(FLAGA(I) .GT. 0) GOTO 11	17610009
10 FLAG(I)(9:9)='I'	17620002
FCOUNT(I)=FCOUNT(I)+1	17630002

```

11   ENDIF                                     17640002
C   FLAG J OCCURS WHEN MDR DAYS TO CLOSE IS ABOVE NORMAL 17650002
    IF (REAL (FLAGJ(I)) .GT. A(JPEER(I),37)+2.0*(A(JPEER(I),38)+.001)) 17660080
      1 THEN                                     17670002
        FLAG(I)(10:10)='J'                     17680002
        FCOUNT(I)=FCOUNT(I)+1                 17690002
      ENDIF                                     17700002
C   FLAG K OCCURS WHEN NET QALI CHANGE IS ABOVE NORMAL 17710002
C   NET QALI IS THE DIFFERENCE BETWEEN QALI RECEIVED AND QALI RESCINDED. 17720002
C   LIMIT ON QALI CHANGED TO +1 SIGMA BASED ON DISCUSSION WITH RICH 17730002
C   ZERILLI AND RON DIPADOVA ON 29 JUNE 87. EXCEPTIONS TO GUIDANCE ARE 17740002
C   GROUPS 26,27,87,127,138,143,148,198,289. 17750002
C   IF (REAL (FLAGK1(I)-FLAGK2(I)).GT. A(JPEER(I),31)+1.0*(A(JPEER(I), 17760060
    IF (REAL (FLAGK1(I)-FLAGK2(I))/(REAL (ONHAND(I))+.01) .GT. 17770027
      1 A(JPEER(I),31)+1.0*(A(JPEER(I),32)+.001)) THEN 17780060
        FLAG(I)(11:11)='K'                     17790002
        FCOUNT(I)=FCOUNT(I)+1                 17800002
      ENDIF                                     17810002
C   FLAG L OCCURS WHEN THE NUMBER OF NONOAR QDR ACTIONS IS ABOVE NORMAL. 17820002
C   LIMIT ON NONOAR CHANGED TO +1 SIGMA BASED ON DISCUSSION WITH RICH 17830002
C   ZERILLI AND RON DIPADOVA ON 29 JUNE 87. NO EXCEPTIONS TO GUIDANCE. 17840002
    IF (REAL (FLAGL(I)) .GT. A(JPEER(I),17)+1.0*(A(JPEER(I),18)+.001)) 17850060
      1 THEN                                     17860002
        FLAG(I)(12:12)='L'                     17870002
        FCOUNT(I)=FCOUNT(I)+1                 17880002
      ENDIF                                     17890002
C   FLAG M OCCURS WHEN THE SYSTEM INDICATOR IS ABOVE NORMAL. 17900002
C   IF (FLAGM(I) .GT. 30.0) THEN 17910024
    IF (FLAGM(I) .GT. A(JPEER(I),33)+2.0*(A(JPEER(I),34))) THEN 17920070
      FLAG(I)(13:13)='M'                       17930002
      FCOUNT(I)=FCOUNT(I)+1                 17940002
    ENDIF                                     17950002
C   FLAG N OCCURS WHEN THE NUMBER OF PCO REQUESTS IS ABOVE NORMAL. 17960002
C   LIMIT ON PCO CHANGED TO +1 SIGMA BASED ON DISCUSSION WITH RICH 17970002
C   ZERILLI AND RON DIPADOVA ON 29 JUNE 87. EXCEPTIONS TO GUIDANCE ARE 17980002
C   GROUPS 125,197. 17990002
    IF (REAL (FLAGN(I)) .GT. A(JPEER(I),9)+1.0*(A(JPEER(I),10)+.001)) 18000060
      1 THEN                                     18010002
        FLAG(I)(14:14)='N'                     18020002
        FCOUNT(I)=FCOUNT(I)+1                 18030002
      ENDIF                                     18040002
C   FLAG O OCCURS WHEN THE NUMBER OF CAD REQUESTS IS ABOVE NORMAL. 18050002
C   LIMIT ON CAD CHANGED TO +1 SIGMA BASED ON DISCUSSION WITH RICH 18060002
C   ZERILLI AND RON DIPADOVA ON 29 JUNE 87. EXCEPTIONS TO GUIDANCE ARE 18070002
C   GROUPS 125,197. 18080002
    IF (REAL (FLAGO(I)) .GT. A(JPEER(I),11)+1.0*(A(JPEER(I),12)+.001)) 18090060
      1 THEN                                     18100002
        FLAG(I)(15:15)='O'                     18110002
        FCOUNT(I)=FCOUNT(I)+1                 18120002
      ENDIF                                     18130002
C   FLAG P OCCURS WHEN THE NUMBER OF ADMIN HOURS IS ABOVE NORMAL. 18140002
    IF (REAL (FLAGP(I)) .GT. A(JPEER(I),1)+2.0*(A(JPEER(I),2)+.001)) 18150071
      1 THEN                                     18160002
        FLAG(I)(16:16)='P'                     18170002
        FCOUNT(I)=FCOUNT(I)+1                 18180002
      ENDIF                                     18190002
C   FLAG Q OCCURS WHEN THERE IS WORK NOT PERFORMED. 18200002
    IF ((FLAGQ1(I)+FLAGQ3(I)) .GT. 0) THEN 18210099
      FLAG(I)(17:17)='Q'                       18220002
      FCOUNT(I)=FCOUNT(I)+1                 18230002
    ENDIF                                     18240002
C   FLAG R OCCURS WHEN NO PE IS PERFORMED AND WORK NOT PERFORMED IS NOT 18250072
C   REPORTED. IT ALSO ERASES THE G FLAG WHEN WORK NOT PERFORMED IS SHOWN. 18260043
    IF (FLAG(I)(7:7) .NE. 'G') GOTO 17 18270043
    IF (FLAGQ3(I) .GT. 0) THEN 18280099
      FLAG(I)(7:7)=' '                         18290043
      FCOUNT(I)=FCOUNT(I)-1                 18300043
    ELSE 18310043
      FLAG(I)(18:18)='R'                       18320080
      FCOUNT(I)=FCOUNT(I)+1                 18330043
    ENDIF                                     18340043
17   RETURN 18350043
    END 18360011
C   18370002
C   *** THIS ROUTINE CHECKS FOR UNUSUAL CORRECTIVE ACTION DISTRIBUTIONS.* 18380002
C   18390002
    SUBROUTINE DISTR(I,FLAGC1,FLAGC2,FLAGC3,FLAGC4,FLAGC5, 18400054
1FCOUNT,JPEER) 18410060
    INTEGER FLAGC1(120),FLAGC2(120),FLAGC3(120),FLAGC4(120), 18420002
1FLAGC5(120),FCOUNT(120),JPEER(120) 18430060

```

CHARACTER FLAG(120)*18,BFSCM(2000)*5	18440099
REAL A(999,38)	18450054
COMMON /RVAR/A	18460002
COMMON /CHTR/BFSCM,FLAG	18470002
IACNT=0	18480002
IBCNT=0	18490002
ICCNT=0	18500002
IDCNT=0	18510002
IECNT=0	18520002
DO 1 K=1,I	18530002
IACNT=IACNT+FLAGC1(K)	18540004
IBCNT=IBCNT+FLAGC2(K)	18550004
ICCNT=ICCNT+FLAGC3(K)	18560004
IDCNT=IDCNT+FLAGC4(K)	18570004
IECNT=IECNT+FLAGC5(K)	18580004
C CHECK FOR UNDER ESCALATION OF CORRECTIVE ACTION	18590002
C SINCE TYPICALLY THE STD DEV OF QDR DATA EXCEEDS THE MEAN, THE	18600002
C DETECTION OF UNDER ESCALATION IS TRIGGERED BY BEING BELOW AVG.	18610002
IF((REAL(IACNT)/REAL(K) .GT. A(JPEER(I),19)+1.0*A(JPEER(I),20))	18620060
1.AND. (REAL(IBCNT)/REAL(K) .LT. A(JPEER(I),21))) GOTO 2	18630060
IF((REAL(IBCNT)/REAL(K) .GT. A(JPEER(I),21)+1.0*A(JPEER(I),22))	18640060
1 AND ((REAL(ICCNT)/REAL(K) .LT. A(JPEER(I),23)) AND.	18650060
2(REAL(IFCNT)/REAL(K) .LT. A(JPEER(I),27))) GOTO 2	18660060
IF(((REAL(ICCNT)/REAL(K) .GT. A(JPEER(I),23)+3.0*A(JPEER(I),24	18670060
1)) .OR. (REAL(IECNT)/REAL(K) .GT. A(JPEER(I),27)+3.0*A(JPEER(I),	18680060
228))) .AND. (REAL(IDCNT)/REAL(K) .LT. A(JPEER(I),25))) GOTO 2	18690060
GO TO 1	18700002
2 FLAG(K)(3:3)='C'	18710002
(K .EQ. I) FCOUNT(I)=FCOUNT(I)+1	18720002
1 C " INUE	18730002
100 RETURN	18740002
END	18750002
C	18760002
C *** THIS SUBROUTINE COMPUTES THE PROCESS EFFECTIVENESS BY DEDUCTING	18770002
C POINTS FOR EACH RED FLAG GENERATED. ALSO THE PROCESS SCORE IS	18780002
C COMBINED WITH THE PRODUCT SCORE TO COMPUTE AN OVERALL SCORE.	18790002
C PRIOR MONTH SCORES ARE NOTED.	18800099
SUBROUTINE SCORER(I,FCOUNT,TOPSCR,TP,JSTRAT)	18810002
INTEGER FCOUNT(120),JSTRAT(120)	18820002
CHARACTER TP(120)*1	18830002
REAL TOPSCR(14,120)	18840099
TOPSCR(10,1)=100.0	18850002
TOPSCR(12,1)=0.0	18860099
TOPSCR(13,1)=0.0	18870099
TOPSCR(14,1)=0.0	18880099
IF(TP(1) .EQ. 'R') THEN	18890002
IF(JSTRAT(1)/5*5 .EQ. JSTRAT(1)) THEN	18900002
C VERY LARGE RESIDENT PENALTY IS 15 PTS PER FLAG EXCEPT FIRST -10 PTS	18910074
IF(FCOUNT(I) .LE. 1) THEN	18920074
TOPSCR(10,1)=100.0-10.0*FCOUNT(I)	18930074
GO TO 10	18940074
ELSE	18950074
TOPSCR(10,1)=105.0-15.0*FCOUNT(I)	18960074
ENDIF	18970076
ELSE	18980076
C MOST RESIDENT PENALTY IS 15 POINTS PER FLAG	18990074
TOPSCR(10,1)=100.0-15.0*FCOUNT(I)	19000076
ENDIF	19010076
ELSE	19020076
C NONRESIDENT PENALTY IS 20 POINTS PER FLAG.	19030074
TOPSCR(10,1)=100.0-20.0*FCOUNT(I)	19040074
10 ENDIF	19050002
IF(TOPSCR(10,1) .LT. 0.0) TOPSCR(10,1)=0.0	19060075
TOPWT=.4	19070002
COAPWT=.6	19080002
TOPSCR(11,1)=TOPSCR(8,1)*TOPWT+TOPSCR(10,1)*COAPWT	19090032
IF(I .EQ. 1) THEN	19100002
GO TO 20	19110099
ELSE	19120099
TOPSCR(12,1)=TOPSCR(8,1-1)	19130099
IF(I .EQ. 2) THEN	19140099
GO TO 20	19150099
ELSE	19160099
TOPSCR(13,1)=TOPSCR(8,1-2)	19170099
IF(I .EQ. 3) THEN	19180099
GO TO 20	19190099
ELSE	19200099
TOPSCR(14,1)=TOPSCR(8,1-3)	19210099
ENDIF	19220099
ENDIF	19230099

20 ENDIF
 RETURN
 END

19240099
19250099
19260002

C THIS MODULE ADDS THE NAME AND CAO CODE TO THE REPORT INPUT RECORD.	00010000
C THIS IS DONE BY MATCHING THE CAGE CODE WITH THE ADRS FILE. THE ADRS	00020000
C FILE MUST BE FLATTENED WITH IEBGENER IN THE PRODUCTION VERSION	00030000
C OF QUEST III.	00040012
C	00050000
CHARACTER FSCM1*5,FSCM2*5,BEFOR*4,AFTER*147,CAO*2,NAMF*20	00060018
(INITIALIZE VARIABLE	00070001
FSCM2='	00080001
C READ A RECORD FROM REPORT INPUT FILE	00090001
10 READ (1,11,END=100) BEFOR,FSCM1,AFTER	00100001
11 FORMAT(A4,A5,A147)	00110018
C COMPARE FSCM'S	00120001
15 IF(FSCM1.EQ.FSCM2) GO TO 40	00130004
IF(FSCM1.LT.FSCM2) THEN	00140008
C IF(LGT(FSCM1,FSCM2)) THEN	00150007
GO TO 30	00160007
ELSE	00170004
GO TO 20	00180007
ENDIF	00190001
C READ A RECORD FROM THE ADRS FILE	00200001
20 READ(2,21,END=10) FSCM2,CAO,NAME	00210001
21 FORMAT(1X,A5,A2,3X,A20,15X)	00220001
GO TO 15	00230001
C ADRS FILE IS AHEAD OF RECORD INPUT FILE	00240001
C READ ANOTHER INPUT RECORD AND DEFAULT	00250001
30 WRITE(3,31) BEFOR,FSCM1,AFTER	00260001
31 FORMAT(A4,A5,A147,'')	00270018
GO TO 10	00280001
C A MATCH HAS BEEN FOUND. APPEND DATA TO INPUT FOR OUTPUT	00290001
40 WRITE(3,41) BEFOR,FSCM1,AFTER,CAO,NAME	00300001
41 FORMAT(A4,A5,A147,A2,A20)	00310018
GO TO 10	00320001
100 STOP	00330001
END	00340001

//GOR6040E JOB (6040,GOR),'GROVER',CLASS=0,MSGCLASS=V	00010099
//RUNFTN EXEC FORTVCG	00030012
//FORT.SYSIN DD DSN=GOR.GROVER.QUEST3(REPORT1),DISP=SHR	00040099
//GO.FT12FOO1 DD DSN=GOR.GROVER.LA.REPZM,DISP=SHR	00050099
//GO.FT13FOO1 DD DSN=GOR.GROVER.Scores.MAY90A,DISP=SHR	00060099
//GO.FT14FOO1 DD DSN=GOR.GROVER.DMINS.LAZM.OCT89,	00070099
// DISP=(NEW,CATLG,DELETE),UNIT=WORKD,	00071099
// SPACE=(TRK,(9,9),RLSE),DCB=(RECFM=FB,LRECL=152,BLKSIZE=15200)	00072099
//*O.FT16FOO1 DD DSN=GOR.GROVER.QUEST.OUTPUT,	00080099
//* DISP=(NEW,CATLG,DELETE),UNIT=TAPE,	00080199
//* DCB=(RECFM=FB,LRECL=133,BLKSIZE=13300),	00080299
//* LABEL=EXPDT=91181	00080399
//GO.FT06FOO1 DD SYSOUT=*	00081099
//SYSOUT DD SYSOUT=*	00090000
//SYSUDUMP DD SYSOUT=*	00100000
//SYSPRINT DD SYSOUT=*	00110000
//	00120000

```

C THIS IS THE QUEST III VERSION OF REPORT GENERATOR TAKEN FROM 00010099
C QUEST II VERSION GOR.GROVER.FOR(DMINRPT4). 00011099
C THIS VERSION IS USED TO PRODUCE A SIMULATED REPORT BASED ON 5A QARCODE 00012099
C THIS IS THE PRODUCTION VERSION TO BE PASSED TO DSAC AND FIELD 00013099
C THE CONCEPT BEHIND THIS VERSION IS TO PASS ALL QUEST II PARAMETERS 00020099
C BUT TO SUPPRESS PRINTING FLAGS, PROGRAM SCORES, ETC. 00030099
C REPORT IS LIMITED TO PRODUCT SCORES. 00040099
C ** THIS PROGRAM GENERATES A DMIN FILE 00060099
C DECLARE AND ARRAY VARIABLES 00070099
  CHARACTER ORGCD(500)*3,ORG*3,FSCM(500)*6,FLAG(500)*18,TYP(500)*1 00080099
  CHARACTER QAR(500)*5,CAO(500)*2,NAME(500)*20,CMDTY(500)*2 00090099
  CHARACTER PVN(500)*1 00091099
  INTEGER MONTH(500),YEAR(500),DEGREE(500),YR,GRP(500) 00100099
  INTEGER ISIZR(500),STARTM,STARTY,ENDMO,ENDYR,NQAR(500) 00110099
  REAL EFF(13,500),REGCUM(11),SECCUM(11,3),BRACUM(11),DIVCUM(11) 00120099
  REAL A(999,2) 00130099
C INITIALIZE COUNTERS AND CUMULATORS 00140099
  CALL INIT(REGCUM,SECCUM,BRACUM,DIVCUM,BRAN,SECT,DIVN) 00150099
C READ IN PEER SCORES 00160099
  I=0 00170099
  II=0 00180099
  2 I=I+1 00190099
C SET DEFAULT VALUES 00191099
  IF(I .LT. 500) THEN 00192099
    A(I,1)=93.53 00200099
    A(I,2)=10.85 00210099
  ELSE 00211099
    A(I,1)=77.97 00212099
    A(I,2)=14.79 00213099
  ENDIF 00214099
  IF(II .GT. I) GOTO 2 00220099
  READ(13,14) II,A(I,1),A(I,2) 00230099
14 FORMAT(17X,I3,25X,F8.4,3X,F8.4,68X) 00240099
  IF(II .GT. I) THEN 00250099
    A(II,1)=A(I,1) 00260099
    A(II,2)=A(I,2) 00270099
    DO 4 K=I,II-1 00280099
      IF(K .LT. 500) THEN 00281099
        A(K,1)=93.53 00282099
        A(K,2)=10.85 00283099
      ELSE 00284099
        A(K,1)=77.97 00285099
        A(K,2)=14.79 00286099
      ENDIF 00287099
    4 CONTINUE 00310099
    I=II 00320099
  ENDIF 00330099
  IF(II .LT. 999) GOTO 2 00340099
C DO 3 J=1,999 00350099
C WRITE(6,14) J,A(J,1),A(J,2) 00360099
C 3 CONTINUE 00370099
C READ FIRST RECORD 00380099
  READ(12,15,END=99) ORGCD(1),FSCM(1),TYP(1),CMDTY(1),PVN(1),NQAR(1) 00390099
  1,MONTH(1),YEAR(1),ISIZR(1),GRP(1),DEGREE(1),QAR(1),FLAG(1), 00400099
  2EFF(1,1),EFF(2,1),EFF(3,1),EFF(4,1),EFF(5,1),EFF(6,1),EFF(7,1), 00410099
  3EFF(8,1),EFF(9,1),EFF(10,1),EFF(11,1),EFF(12,1),EFF(13,1), 00420099
  4STARTM,STARTY,ENDMO,ENDYR,CAO(1),NAME(1) 00421099
15 FORMAT(A3,A6,A1,A2,A1,3I3,2I4,I2,A5,A18,13F7.1,4I2,2X,A2,A20) 00422099
  DO 13 J=1,11 00430099
    IF(TYP(1) .EQ. 'R') THEN 00440099
      IF(ISIZR(1) .EQ. 385) ISIZR(1)=391 00450099
      IRESCT=(MOD((ISIZR(1)+4),5)+1)**2 00460099
      SECCUM(J,1)=EFF(J,1)*REAL(IRESCT) 00470099
      NRESCT=0 00480099
    ELSE 00490099
      SECCUM(J,2)=EFF(J,1) 00500099
      NRESCT=1 00510099
      IRESCT=0 00520099
    ENDIF 00530099
  13 CONTINUE 00540099
C READ REMAINING RECORDS FOR SECTION 00550099
  11 LIN=0 00560099
  DO 20 I=2,500 00570099
    READ(12,16,END=99) ORGCD(I),FSCM(I),TYP(I),CMDTY(I),PVN(I),NQAR(I) 00611099
    1,MONTH(I),YEAR(I),ISIZR(I),GRP(I),DEGREE(I),QAR(I),FLAG(I), 00612099
    2EFF(1,I),EFF(2,I),EFF(3,I),EFF(4,I),EFF(5,I),EFF(6,I),EFF(7,I), 00613099
    3EFF(8,I),EFF(9,I),EFF(10,I),EFF(11,I),EFF(12,I),EFF(13,I), 00614099
    4CAO(I),NAME(I) 00615099
  20 FORMAT(A3,A6,A1,A2,A1,3I3,2I4,I2,A5,A18,13F7.1,10X,A2,A20) 00620099

```

C *** IF A NEW MONTH IS FOUND, UPDATE ALL SUMMARY STATS.	00622099
IF(MONTH(I) .NE. MONTH(I-1)) THEN	00640099
101 CALL HEADR(I,MONTH,YEAR,ORGCD,CAO)	00650099
CALL WRITER(I,LIN,FSCM,GRP,DEGREE,FLAG,EFF,MONTH,YEAR,ORGCD,A,	00660099
1TYP,QAR,CAO,NAME,CMDTY,PVN,STARTM,STARTY,NQAR)	00670099
CALL SECTOT(I,SECCUM,IRESCT,NRESCT,EFF,ORGCD,FSCM,YEAR,MONTH,	00680099
1GRP,DEGREE,FLAG,BRACUM,TYP,SECT,ORG,MON,YR,CAO,NAME,	00690099
2ISIZR,QAR,CMDTY,PVN,NQAR)	00700099
CALL BRATOT(BRACUM,BRAN,SECT,DIVCUM,ORG,MON,YR)	00710099
CALL DIVTOT(DIVCUM,DIVN,BRAN,REGCUM,ORG,MON,YR)	00720099
CALL REGTOT(REGCUM,DIVN,MON,YR)	00730099
IF(IENDR .EQ. 1) GOTO 100	00740099
GO TO 11	00750099
ENDIF	00760099
C *** IF A NEW DIVISION IS FOUND, UPDATE ALL SUMMARY STATS.	00770099
IF(ORGCD(I)(1:1) .NE. ORGCD(I-1)(1:1)) THEN	00780099
CALL HEADR(I,MONTH,YEAR,ORGCD,CAO)	00790099
CALL WRITER(I,LIN,FSCM,GRP,DEGREE,FLAG,EFF,MONTH,YEAR,ORGCD,A,	00800099
1TYP,QAR,CAO,NAME,CMDTY,PVN,STARTM,STARTY,NQAR)	00810099
CALL SECTOT(I,SECCUM,IRESCT,NRESCT,EFF,ORGCD,FSCM,YEAR,MONTH,	00820099
1GRP,DEGREE,FLAG,BRACUM,TYP,SECT,ORG,MON,YR,CAO,NAME,	00830099
2ISIZR,QAR,CMDTY,PVN,NQAR)	00840099
CALL BRATOT(BRACUM,BRAN,SECT,DIVCUM,ORG,MON,YR)	00850099
CALL DIVTOT(DIVCUM,DIVN,BRAN,REGCUM,ORG,MON,YR)	00860099
GO TO 11	00870099
ENDIF	00880099
C *** IF A NEW BRANCH IS FOUND, UPDATE ALL SUMMARY STATS.	00890099
IF(ORGCD(I)(1:2) .NE. ORGCD(I-1)(1:2)) THEN	00900099
CALL HEADR(I,MONTH,YEAR,ORGCD,CAO)	00910099
CALL WRITER(I,LIN,FSCM,GRP,DEGREE,FLAG,EFF,MONTH,YEAR,ORGCD,A,	00920099
1TYP,QAR,CAO,NAME,CMDTY,PVN,STARTM,STARTY,NQAR)	00930099
CALL SECTOT(I,SECCUM,IRESCT,NRESCT,EFF,ORGCD,FSCM,YEAR,MONTH,	00940099
1GRP,DEGREE,FLAG,BRACUM,TYP,SECT,ORG,MON,YR,CAO,NAME,	00950099
2ISIZR,QAR,CMDTY,PVN,NQAR)	00960099
CALL BRATOT(BRACUM,BRAN,SECT,DIVCUM,ORG,MON,YR)	00970099
GO TO 11	00980099
ENDIF	00990099
C *** IF A NEW SECTION IS FOUND, UPDATE ALL SUMMARY STATS.	01000099
IF(ORGCD(I)(1:3) .NE. ORGCD(I-1)(1:3)) THEN	01010099
CALL HEADR(I,MONTH,YEAR,ORGCD,CAO)	01020099
CALL WRITER(I,LIN,FSCM,GRP,DEGREE,FLAG,EFF,MONTH,YEAR,ORGCD,A,	01030099
1TYP,QAR,CAO,NAME,CMDTY,PVN,STARTM,STARTY,NQAR)	01040099
CALL SECTOT(I,SECCUM,IRESCT,NRESCT,EFF,ORGCD,FSCM,YEAR,MONTH,	01050099
1GRP,DEGREE,FLAG,BRACUM,TYP,SECT,ORG,MON,YR,CAO,NAME,	01060099
2ISIZR,QAR,CMDTY,PVN,NQAR)	01070099
GO TO 11	01080099
C CURRENT RECORD IS IN THE SAME SECTION. UPDATE SECTION COUNTERS	01090099
ELSE	01100099
IF(TYP(I) .EQ. 'R') THEN	01120099
IF(ISIZR(I) .EQ. 625) ISIZR(I)=991	01130099
IRESCT=IRESCT+(MOD((ISIZR(I)+4),5)+1)**2	01140099
DO 23 J=1,11	01150099
SECCUM(J,1)=SECCUM(J,1)+EFF(J,I)*(REAL(MOD((ISIZR(I)+4),5)+1))	01160099
1 **2	01170099
23 CONTINUE	01180099
ELSE	01190099
NRESCT=NRESCT+1	01200099
DO 24 J=1,11	01210099
SECCUM(J,2)=SECCUM(J,2)+EFF(J,I)	01220099
24 CONTINUE	01230099
ENDIF	01240099
ENDIF	01250099
20 CONTINUE	01260099
98 WRITE(6,97)	01270099
97 FORMAT(1X,'TOO MANY FACILITIES ASSIGNED TO SECTION.ABNORMAL END')	01280099
C NORMAL END . FINISH PROCESSING LAST ORGS AND END JOB.	01290099
99 IENDR=1	01300099
GO TO 101	01310099
100 STOP	01320099
END	01330099
SUBROUTINE INIT(REGCUM,SECCUM,BRACUM,DIVCUM,BRAN,SECT,DIVN)	01340099
REAL SECCUM(11,3),BRACUM(11),REGCUM(11),DIVCUM(11)	01350099
DO 10 I=1,11	01360099
REGCUM(I)=0.0	01370099
SECCUM(I,1)=0.0	01380099
SECCUM(I,2)=0.0	01390099
SECCUM(I,3)=0.0	01400099
BRACUM(I)=0.0	01410099
DIVCUM(I)=0.0	01420099
10 CONTINUE	01430099

```

BRAN=0.0                                01440099
SECT=0.0                                01450099
DIVN=0.0                                01460099
RETURN                                  01470099
END                                      01480099
SUBROUTINE HEADR(I,MONTH,YEAR,ORGCD,CAO) 01490099
  INTEGER MONTH(500),YEAR(500)          01500099
  CHARACTER ORGCD(500)*3,CAO(500)*2     01510099
  *** NEXT LINE SUPPRESSED ALL BUT FINAL MONTH HEADER 01510199
    WRITE(C,17) MONTH(I-1),YEAR(I-1),ORGCD(I-1),CAO(I-1) 01520099
  17 FORMAT('1',2I3,' QUEST REPORT FOR SECTION ',A3,5X,A2/ 01571099
    150X,'CONTRACTOR EFFECTIVENESS',27X,'FIRST',' SECOND',' THIRD'/ 01571199
    1 ' ',2X,'CAGE',5X,'NAME QAR COMM/ QAS',2X,'CA',5X,'PODR', 01572099
    2 5X,'PA',5X,'WVRS',5X,'MRB',5X,'DEVN',5X,'ECP',5X,'TOTAL',3X, 01573099
    3 'PRIOR PRIOR PRIOR',5X,'PEER'/' ',26X,'PVN ',62X,'SCORE', 01574099
    4 3X,'MONTH',3X,'MONTH',3X,'MONTH',4X,'GRP/RTG') 01575099
  10 RETURN                               01580099
  END                                     01590099
  SUBROUTINE WRITER(I,LIN,FSCM,GRP,DEGREE,FLAG,EFF,MONTH,YEAR,ORGCD, 01600099
  1A,TYPE,QARCD,CAO,NAME,CMDTY,PVN,STARTM,STARTY,NQAR) 01610099
    INTEGER DEGREE(500),MONTH(500),YEAR(500),GRP(500),STARTM,STARTY 01620099
    INTEGER NQAR(500) 01630099
    CHARACTER FLAG(500)*18,FSCM(500)*6,ORGCD(500)*3,RANKIT*1 01640099
    CHARACTER TYPE(500)*1,QARCD(500)*5,CAO(500)*2,NAME(500)*20 01650099
    CHARACTER CMDTY(500)*2,PVN(500)*1 01651099
    REAL EFF(13,500),A(999,2) 01660099
    LIN=0 01670099
    NUMLIN=1 01680099
    KPG=0 01690099
  10 DO 18 KK=1,60 01700099
  C 10 DO 18 KK=1,50 01710099
    K=KPG*50+KK 01720099
    LIN=LIN+1 01730099
    IF(LIN .GE. NUMLIN) GOTO 22 01740099
    IF(K .GT. 1) THEN 01741099
      IF((DEGREE(K-1) .LT. 3) .AND. (DEGREE(K) .GE. 3)) THEN 01742099
        WRITE(6,15) 01743099
  15 FORMAT(' ') 01744099
        ENDIF 01745099
        ENDIF 01746099
        CALL PEER(K,GRP,EFF,A,RANKIT) 01800099
        IF(DEGREE(K)/2*2 .NE. DEGREE(K)) FSCM(K)(1:1)='*' 01800199
        IF((MONTH(K) .EQ. STARTM) .AND. (YEAR(K) .EQ. STARTY)) THEN 01801099
          WRITE(6,23) FSCM(K),NAME(K)(1:10),QARCD(K),CMDTY(K),PVN(K), 01801199
          2NQAR(K),EFF(7,K),EFF(8,K),EFF(2,K),EFF(5,K),EFF(4,K),EFF(3,K), 01801399
          3EFF(6,K),EFF(9,K),GRP(K),RANKIT 01801499
  23 FORMAT(1X,A6,1X,A10,1X,A5,1X,A2,1X,A1,1X,I3,2X,7(F5.1,3X),2X, 01801599
          1 F5.1,27X,I4,2X,A1) 01801699
          GO TO 25 01802099
        ENDIF 01803099
        IF(((MONTH(K) .EQ. STARTM+1) .AND. (YEAR(K) .EQ. STARTY)) .OR. 01804099
          1 ((MONTH(K) .EQ. STARTM-1) .AND. (YEAR(K) .EQ. STARTY+1))) THEN 01804199
          WRITE(6,21) FSCM(K),NAME(K)(1:10),QARCD(K),CMDTY(K),PVN(K), 01805099
          2NQAR(K),EFF(7,K),EFF(8,K),EFF(2,K),EFF(5,K),EFF(4,K),EFF(3,K), 01806099
          3EFF(6,K),EFF(9,K),EFF(11,K),GRP(K),RANKIT 01807099
  21 FORMAT(1X,A6,1X,A10,1X,A5,1X,A2,1X,A1,1X,I3,2X,7(F5.1,3X),2X, 01808099
          1 F5.1,3X,F5.1,19X,I4,2X,A1) 01809099
          GO TO 25 01810099
        ENDIF 01820099
        IF(((MONTH(K) .EQ. STARTM+2) .AND. (YEAR(K) .EQ. STARTY)) .OR. 01830099
          1 ((MONTH(K) .EQ. STARTM-10) .AND. (YEAR(K) .EQ. STARTY+1))) THEN 01840099
          WRITE(6,20) FSCM(K),NAME(K)(1:10),QARCD(K),CMDTY(K),PVN(K), 01850099
          2NQAR(K),EFF(7,K),EFF(8,K),EFF(2,K),EFF(5,K),EFF(4,K),EFF(3,K), 01860099
          3EFF(6,K),EFF(9,K),EFF(11,K),EFF(12,K),GRP(K),RANKIT 01870099
  20 FORMAT(1X,A6,1X,A10,1X,A5,1X,A2,1X,A1,1X,I3,2X,7(F5.1,3X),2X, 01880099
          1 F5.1,3X,F5.1,3X,F5.1,11X,I4,2X,A1) 01890099
          GO TO 25 01900099
        ENDIF 01910099
        WRITE(6,24) FSCM(K),NAME(K)(1:10),QARCD(K),CMDTY(K),PVN(K), 02041099
        1NQAR(K),EFF(7,K),EFF(8,K),EFF(2,K), 02042099
        2EFF(5,K),EFF(4,K),EFF(3,K),EFF(6,K),EFF(9,K),EFF(11,K),EFF(12,K), 02042199
        3EFF(13,K),GRP(K),RANKIT 02043099
  24 FORMAT(1X,A6,1X,A10,1X,A5,1X,A2,1X,A1,1X,I3,2X,7(F5.1,3X),2X, 02044099
        14(F5.1,3X),I4,2X,A1) 02045099
  25 WRITE(14,27) ORGCD(K),FSCM(K)(2:6),FLAG(K),EFF(1,K),EFF(2,K), 02050099
        1EFF(3,K),EFF(4,K),EFF(5,K),EFF(6,K),EFF(7,K),EFF(8,K),EFF(9,K), 02060099
        2EFF(10,K),EFF(11,K),RANKIT,FSCM(K)(1:1),YEAR(K),MONTH(K),TYPE(K), 02070099
        3QARCD(K),NAME(K),GRP(K),CAO(K),NQAR(K),CMDTY(K),PVN(K) 02080099
  27 FORMAT(A3,' ',A5,' ',A18,' ',3(F5.1,' '),8(F5.1,' '), 02090099
        12(A1,' ').2(I2,' '),5,' ',A1,' ',A5,' ',A20,' ',I4,' ',A2,' ',I302100099

```

2, '1', A2, '1', A1)	02110099
18 CONTINUE	02130099
CALL HEADR(I, MONTH, YEAR, ORGCD, CAO)	02140099
KPG=KPG+1	02150099
GO TO 10	02160099
22 RETURN	02170099
END	02180099
SUBROUTINE PEER(K, GRP, EFF, A, RANKIT)	02190099
INTEGER GRP(500)	02200099
CHARACTER RANKIT*1	02210099
REAL EFF(13,500), A(999,2)	02220099
IF ((EFF(9,K) .GT. A(GRP(K),1)+A(GRP(K),2)) .OR. (EFF(9,K) .GE.	02230099
1 100.0)) THEN	02231099
RANKIT='A'	02240099
GO TO 10	02250099
ENDIF	02260099
IF (GRP(K) .GT. 500) THEN	02261099
IF (EFF(9,K) .GT. A(GRP(K),1)+.5*A(GRP(K),2)) THEN	02270099
RANKIT='B'	02280099
GO TO 10	02290099
ENDIF	02300099
IF (EFF(9,K) .GT. A(GRP(K),1)-.5*A(GRP(K),2)) THEN	02310099
RANKIT='C'	02320099
GO TO 10	02330099
ENDIF	02340099
IF (EFF(9,K) .GT. A(GRP(K),1)-A(GRP(K),2)) THEN	02350099
RANKIT='D'	02360099
GO TO 10	02370099
ENDIF	02380099
RANKIT='F'	02390099
GO TO 10	02390199
ELSE	02391099
IF (EFF(9,K) .GT. A(GRP(K),1)) THEN	02392099
RANKIT='B'	02393099
GO TO 10	02394099
ENDIF	02395099
IF (EFF(9,K) .GT. A(GRP(K),1)-1.0*A(GRP(K),2)) THEN	02396099
RANKIT='C'	02397099
GO TO 10	02398099
ENDIF	02399099
IF (EFF(9,K) .GT. A(GRP(K),1)-2.0*A(GRP(K),2)) THEN	02399199
RANKIT='D'	02399299
GO TO 10	02399399
ENDIF	02399499
RANKIT='F'	02399599
GO TO 10	02399699
ENDIF	02399799
10 RETURN	02400099
END	02410099
SUBROUTINE SECTOT(I, SECCUM, IRESCT, NRESCT, EFF, ORGCD, FSCM, YEAR,	02420099
1 MONTH, GRP, DEGREE, FLAG, BRACUM, TYP, SECT, ORG, MON, YR, CAO, NAME,	02430099
2 ISIZR, QAR, CMDTY, PVN, NQAR)	02440099
CHARACTER ORGCD(500)*3, FSCM(500)*6, FLAG(500)*18, TYP(500)*1, ORG*3	02450099
CHARACTER CAO(500)*2, NAME(500)*20, QAR(500)*5, CMDTY(500)*2	02460099
CHARACTER PVN(500)*1	02461099
INTEGER MONTH(500), YEAR(500), DEGREE(500), YR, GRP(500), ISIZR(500)	02470099
INTEGER NQAR(500)	02480099
REAL EFF(13,500), SECCUM(11,3), BRACUM(11), DIVCUM(11), RLGCUM(11)	02490099
REAL NMRTR, DNMRTR, RSWGT	02500099
SECT=SECT+REAL(IRESCT)+REAL(NRESCT)/5.0	02510099
RSWGT=.2	02520099
DO 1 J=1, 11	02530099
NMRTR=(SECCUM(J,1)+RSWGT*SECCUM(J,2))	02540099
DNMRTR=(REAL(IRESCT)+RSWGT*REAL(NRESCT)+.00001)	02550099
SECCUM(J,3)=NMRTR/DNMRTR	02560099
SECCUM(J,1)=SECCUM(J,1)/(REAL(IRESCT)+.00001)	02570099
SECCUM(J,2)=SECCUM(J,2)/(REAL(NRESCT)+.00001)	02580099
BRACUM(J)=BRACUM(J)+SECCUM(J,3)*(REAL(IRESCT)+REAL(NRESCT)/5.0)	02590099
1 CONTINUE	02600099
WRITE(6,5) ORGCD(I-1), SECCUM(7,1), SECCUM(8,1), SECCUM(2,1),	02651099
1 SECCUM(5,1), SECCUM(4,1), SECCUM(3,1), SECCUM(6,1), SECCUM(9,1)	02652099
5 FORMAT(' ', 13('-----'), '---/' ' ', ' WTD AVG ', 3X, A3/' ',	02654099
1 ' RESIDENT', 21X, 7(F5.1, 3X), 2X, F5.1)	02655099
WRITE(6,11) SECCUM(7,2), SECCUM(8,2), SECCUM(2,2), SECCUM(5,2),	02691099
1 SECCUM(4,2), SECCUM(3,2), SECCUM(6,2), SECCUM(9,2)	02692099
11 FORMAT(' ', 5X, 'NONRESIDENT', 18X, 7(F5.1, 3X), 2X, F5.1)	02694099
WRITE(6,12) SECCUM(7,3), SECCUM(8,3), SECCUM(2,3), SECCUM(5,3),	02731099
1 SECCUM(4,3), SECCUM(3,3), SECCUM(6,3), SECCUM(9,3)	02732099
12 FORMAT(' ', 5X, 'COMBINED ', 18X, 7(F5.1, 3X), 2X, F5.1)	02734099
100 WRITE(14,7) ORGCD(I-1), SECCUM(1,1), SECCUM(2,1), SECCUM(3,1),	02740099

```

1SECCUM(4,1),SECCUM(5,1),SECCUM(6,1),SECCUM(7,1),SECCUM(8,1),
2SECCUM(9,1),SECCUM(10,1),SECCUM(11,1),YEAR(I-1),MONTH(I-1)
7 FORMAT(A3,'1RES 1SECTION SUMMARY 1',11(F5.1,'1'),' 1 1',
12(I2,'1'),'41 ')
WRITE(14,8) ORGCD(I-1),SECCUM(1,2),SECCUM(2,2),SECCUM(3,2),
1SECCUM(4,2),SECCUM(5,2),SECCUM(6,2),SECCUM(7,2),SECCUM(8,2),
2SECCUM(9,2),SECCUM(10,2),SECCUM(11,2),YEAR(I-1),MONTH(I-1)
8 FORMAT(A3,'1N/R 1SECTION SUMMARY 1',11(F5.1,'1'),' 1 1',
12(I2,'1'),'41 ')
WRITE(14,9) ORGCD(I-1),SECCUM(1,3),SECCUM(2,3),SECCUM(3,3),
1SECCUM(4,3),SECCUM(5,3),SECCUM(6,3),SECCUM(7,3),SECCUM(8,3),
2SECCUM(9,3),SECCUM(10,3),SECCUM(11,3),YEAR(I-1),MONTH(I-1)
9 FORMAT(A3,'1AGG 1SECTION SUMMARY 1',11(F5.1,'1'),' 1 1',
12(I2,'1'),'41 ')
ORG=ORGCD(I-1)
MON=MONTH(I-1)
YR=YEAR(I-1)
ORGCD(1)=ORGCD(I)
FSCM(1)=FSCM(I)
YEAR(1)=YEAR(I)
MONTH(1)=MONTH(I)
TYP(1)=TYP(I)
QAR(1)=QAR(I)
CMDIY(1)=CMDIY(I)
PVN(1)=PVN(I)
NQAR(1)=NQAR(I)
ISIZR(1)=ISIZR(I)
GRP(1)=GRP(I)
DEGREE(1)=DEGREE(I)
FLAG(1)=FLAG(I)
CAO(1)=CAO(I)
NAME(1)=NAME(I)
EFF(12,1)=EFF(12,I)
EFF(13,1)=EFF(13,I)
DO 20 J=1,11
EFF(J,1)=EFF(J,I)
IF (TYP(1) .EQ. 'R') THEN
IF (ISIZR(1) .EQ. 625) GRP(1)=991
IRESCT=(MOD((ISIZR(1)+4),5)+1)**2
NRESCT=0
SECCUM(J,1)=EFF(J,1)*REAL(IRESCT)
SECCUM(J,2)=0
ELSE
IRESCT=0
NRESCT=1
SECCUM(J,1)=0
SECCUM(J,2)=EFF(J,1)
ENDIF
20 CONTINUE
RETURN
END
SUBROUTINE BRATOT(BRACUM,BRAN,SECT,DIVCUM,ORG,MON,YR)
REAL BRACUM(11),DIVCUM(11)
INTEGER YR
CHARACTER ORG*3
DO 10 K=1,11
DIVCUM(K)=DIVCUM(K)+BRACUM(K)
BRACUM(K)=BRACUM(K)/SECT
10 CONTINUE
WRITE(6,15) ORG(1:2)
15 FORMAT(' ',13('-----'),'---/' ' ' WTD AVG',',',2X,A2)
WRITE(6,40) BRACUM(7),BRACUM(8),BRACUM(2),BRACUM(5),
1BRACUM(4),BRACUM(3),BRACUM(6),BRACUM(9)
40 FORMAT('+',34X,7(F5.1,3X),2X,F5.1)
50 WRITE(14,41) ORG(1:2),BRACUM(1),BRACUM(2),BRACUM(3),BRACUM(4),
1BRACUM(5),BRACUM(6),BRACUM(7),BRACUM(8),BRACUM(9),BRACUM(10),
2BRACUM(11),YR,MON
41 FORMAT(A2,' 1BP 1',18X,' 1',11(F5.1,'1'),' 1 1',2(I2,'1'),'31 ')
DO 16 J=1,11
BRACUM(J)=0.0
16 CONTINUE
BRAN=BRAN+SECT
SECT=0.0
RETURN
END
SUBROUTINE DIVTOT(DIVCUM,DIVN,BRAN,REGCUM,ORG,MON,YR)
REAL DIVCUM(11),REGCUM(11)
INTEGER YR
CHARACTER ORG*3
DO 10 K=1,11

```

```

02750099
02760099
02770099
02780099
02790099
02800099
02810099
02820099
02830099
02840099
02850099
02860099
02870099
02880099
02890099
02900099
02910099
02920099
02930099
02940099
02950099
02960099
02961099
02962099
02963099
02964099
02970099
02980099
02990099
03000099
03010099
03020099
03041099
03042099
03050099
03060099
03070099
03080099
03090099
03100099
03110099
03120099
03130099
03140099
03150099
03160099
03170099
03180099
03190099
03200099
03210099
03220099
03230099
03240099
03250099
03260099
03270099
03280099
03290099
03300099
03310099
03341099
03342099
03343099
03350099
03360099
03370099
03380099
03390099
03400099
03410099
03420099
03430099
03440099
03450099
03460099
03470099
03480099
03490099
03500099

```


REGCUM(K)=REGCUM(K)+DIVCUM(K)	03510099
DIVCUM(K)=DIVCUM(K)/BRAN	03520099
10 CONTINUE	03530099
WRITE(6,35) ORG(1:1)	03540099
35 FORMAT(' ',13('-----'),'---/' ' ',WTD AVG',',2X,A1)	03550099
WRITE(6,40) DIVCUM(7),DIVCUM(8),DIVCUM(2),DIVCUM(5),	03581099
1DIVCUM(4),DIVCUM(3),DIVCUM(6),DIVCUM(9)	03582099
40 FORMAT('+',34X,7(F5.1,3X),2X,F5.1)	03583099
50 WRITE(14,60) ORG(1:1),DIVCUM(1),DIVCUM(2),DIVCUM(3),DIVCUM(4),	03590099
1DIVCUM(5),DIVCUM(6),DIVCUM(7),DIVCUM(8),DIVCUM(9),DIVCUM(10),	03600099
2DIVCUM(11),YR,MON	03610099
60 FORMAT(A1,' 1DIV 1',18X,'1',11(F5.1,'1'),' 1 1',2(I2,'1'),'21')	03620099
DO 36 J=1,11	03630099
DIVCUM(J)=0.0	03640099
36 CONTINUE	03650099
DIVN=DIVN+BRAN	03660099
BRAN=0.0	03670099
RETURN	03680099
END	03690099
SUBROUTINE REGTOT(REGCUM,DIVN,MON,YR)	03700099
REAL REGCUM(11)	03710099
INTEGER YR	03720099
DO 10 K=1,11	03730099
REGCUM(K)=REGCUM(K)/DIVN	03740099
10 CONTINUE	03750099
WRITE(6,45) MON,YR	03760099
45 FORMAT(' ',13('-----'),'---/' ' ',REGION SUMMARY ',3X,2I3)	03770099
WRITE(6,40) REGCUM(7),REGCUM(8),REGCUM(2),REGCUM(5),	03801099
1REGCUM(4),REGCUM(3),REGCUM(6),REGCUM(9)	03802099
40 FORMAT('+',34X,7(F5.1,3X),2X,F5.1)	03803099
50 WRITE(14,41) REGCUM(1),REGCUM(2),REGCUM(3),REGCUM(4),REGCUM(5),	03810099
1REGCUM(6),REGCUM(7),REGCUM(8),REGCUM(9),REGCUM(10),REGCUM(11),	03820099
2YR,MON	03830099
41 FORMAT('---1 1REGION ROLLUP 1',11(F5.1,'1'),' 1 1',	03840099
12(I2,'1'),'11')	03850099
DO 46 J=1,11	03860099
REGCUM(J)=0.0	03870099
46 CONTINUE	03880099
DIVN=0.0	03890099
RETURN	03900099
END	03910099

APPENDIX C

References

1. Grover, Paul E., Analysis of Quality Assurance (QA) Effectiveness, DLA-LO Project 3071, July 1988.
2. Hwang, Ching-Lai and Lin, Ming-Jeng, Group Decision Making Under Multiple Criteria, Springer-Verlag, Berlin Heidelberg, 1987.
3. Hwang, Ching-Lai and Yoon, Kwangsun, Multiple Attribute Decision Making, Springer-Verlag, Berlin Heidelberg and New York, 1981.
4. Grover, Paul E., Quality Effectiveness Sensing Technique (QUEST) Release II, DLA-LO Project DLA-90-P90271, March 1990.
5. Freund, John E., Modern Elementary Statistics, Prentice-Hall, Third Edition, Englewood Cliffs, NJ, 1967.

REPORT DOCUMENTATION PAGE			Form Approved OMB No. 0704-0188	
<small>Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information, and comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to: Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204 Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188) Washington, DC 20503</small>				
1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE March 1991	3. REPORT TYPE AND DATES COVERED Final		
4. TITLE AND SUBTITLE Quality Effectiveness Sensing Technique Release 3.0			5. FUNDING NUMBERS	
6. AUTHOR(S) Paul E. Grover				
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) HQ Defense Logistics Agency Operations Research and Economic Analysis Office (DLA-LO) Cameron Station Alexandria, VA 22304-6100			8. PERFORMING ORGANIZATION REPORT NUMBER DLA-91-P90272	
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) Defense Logistics Agency Cameron Station Alexandria, VA 22304-6100			10. SPONSORING/MONITORING AGENCY REPORT NUMBER	
11. SUPPLEMENTARY NOTES				
12a. DISTRIBUTION/AVAILABILITY STATEMENT Public Release; Unlimited Distribution			12b. DISTRIBUTION CODE	
13. ABSTRACT (Maximum 200 words) <p>This report documents the logic, structure and code for QUEST Release 3.0. QUEST Release 3.0 measures the effectiveness of the contractor's QA operations by comparing the contractor's key indicators with those of similar contractors (or peers). Based on those peer comparisons and trends, an effectiveness score for each indicator is computed and a weighted average of all indicators produces a bottom line effectiveness rating. Release 3.0 was validated by incorporating the profound knowledge of experts in the weighting factors and program logic. A study advisory group, consisting of field and Headquarters DLA QA personnel guided the effort. A statistical test was passed which compared QUEST Release 3.0 results with expert opinion on certain contractors. Because of the track record of earlier releases, the use of knowledgeable experts in the development process and the results of validity tests, it is concluded that QUEST 3.0 is implementable, valid and meets the objective of measuring contractor QA effectiveness. It is recommended that Release 3.0 be implemented throughout the Defense Contract Management Command (DCMC).</p>				
14. SUBJECT TERMS Quality, Effectiveness, Contractor			15. NUMBER OF PAGES 61	
			16. PRICE CODE	
17. SECURITY CLASSIFICATION OF REPORT UNCLASSIFIED	18. SECURITY CLASSIFICATION OF THIS PAGE UNCLASSIFIED	19. SECURITY CLASSIFICATION OF ABSTRACT UNCLASSIFIED	20. LIMITATION OF ABSTRACT	